Organización
de las Naciones Unidas
para la Educación,
la Ciencia y la Cultura

Oficina Regional de Ciencia
para América Latina y el Caribe

Sector Comunicación e Información
UNESCO Montevideo

GUÍA PRÁCTICA SOBRE SOFTWARE LIBRE

SU SELECCIÓN Y APLICACIÓN LOCAL EN AMÉRICA LATINA Y EL CARIBE



Fernando da Rosa
Federico Heinz

Organización
de las Naciones Unidas
para la Educación,
la Ciencia y la Cultura

Oficina Regional de Ciencia
para América Latina y el Caribe

Sector Comunicación e Información
UNESCO Montevideo



GUÍA PRÁCTICA SOBRE SOFTWARE LIBRE

SU SELECCIÓN Y APLICACIÓN LOCAL EN AMÉRICA LATINA Y EL CARIBE

Fernando da Rosa
Federico Heinz

UNESCO
Coordinador: Günther Cyranek
Autores: Fernando da Rosa y Federico Heinz
Otros colaboradores: Beatriz Busaniche y Gustavo Boksar
Diseño de Portada e Imagen: Jorge Ambiado
Crédito de las fotos: Foto de Linus Torvalds, Tomado de la Wikipedia bajo licencia GNU Free Documentation License con permiso de Martin Streicher, Editor en Jefe de LINUXMAG.
Foto de Richard Stallman, Fotografía de Bill Ebbesen, licenciada como Dominio Público.
Fotos de Latinoware 2006, LACFREE 2005, Jornadas Regionales 2004. Fotografías de Fernando da Rosa
Mapa Conceptual del Software Libre: realizado originalmente por René Mérou, bajo licencia GPL. Modificado por Fernando da Rosa, Jorge Ambiado y Gustavo Boksar

Por más información, por favor diríjase a:
Günther Cyranek
Sector de Comunicación e Información
Oficina Regional de Ciencia de la UNESCO para América Latina y el Caribe
Edificio MERCOSUR
Dr. Luis Piera 1992
11200 Montevideo, Uruguay
Teléfono: (+598 2) 413 20 75 ext. 126 / 124
Fax: (+598 2) 413 20 94
Correo electrónico: g.cyranek@unesco.org
Oficina Regional de Ciencia de la UNESCO para América Latina y el Caribe

ISBN 92-9089-103-3
Guía práctica sobre Software Libre. Su selección y aplicación local en América Latina y el Caribe.
Fernando da Rosa y Federico Heinz.
Paginado, impreso y encuadernado en: Mastergraf SRL - Gral. Pagola 1727 - CP 11800 - Tel.: 203 4760
Montevideo - Uruguay
Depósito legal 339.016 - Comisión del Papel
Edición amparada al Decreto 218/96

AGRADECIMIENTOS

Los autores quisieran agradecer a todos aquellos que han contribuido con esta guía, proporcionado información, opiniones y consejo. Un agradecimiento especial por cierto, a Richard Stallman, la Free Software Foundation y el proyecto GNU, por su incansable y visionaria labor de dos décadas, sin la cual estaríamos muy lejos de una comprensión cabal del fenómeno del software, de las consecuencias de sus modos de licenciamiento, y de la concreción del software libre como una realidad cotidiana.

Esta guía no hubiera sido posible sin la colaboración, en la revisión del material, crítica constructiva y aportes de Beatriz Busaniche y Gustavo Boksar.

También deseamos agradecer los aportes siempre valiosos de Rodolfo Pilas y en general a toda la comunidad del Software Libre, la cual construye su reflexión en torno al Software Libre, de la misma manera en que lo desarrolla, aportando sobre la base de lo realizado anteriormente, en una dinámica que hace del movimiento del Software Libre algo en permanente desarrollo.

REPRODUCCIÓN

SOFTWARE LIBREMENTE

En 2001, al comenzar UNESCO a brindar apoyo al movimiento de software libre, se cumplieron 18 años desde que lanzamos el movimiento y empezamos a desarrollar el sistema operativo GNU. Hemos recorrido un largo camino.

En 1984 era imposible usar un sistema moderno de computadora sin instalar un sistema operativo propietario, que hubiera sido entonces obtenido bajo una licencia de uso restrictiva. Nadie tenía permiso para compartir libremente el software con otros usuarios y prácticamente todos los usuarios tenían bloqueado el acceso que les hubiera permitido ajustar el software a sus propias necesidades. Los propietarios del software habían levantado paredes para dividirnos.

El Proyecto GNU fue fundado para cambiar esto. Su primera meta: desarrollar un sistema operativo portable compatible con Unix que fuera hecho 100% de software libre. Ni 95%, ni 99.5%, sino 100%, de manera que los usuarios fueran libres de distribuir el sistema completo y de cambiar y contribuir con cualquier parte de él. El nombre del sistema, GNU, es un acrónimo recursivo que significa "GNU no es Unix"-- como manera de rendir homenaje a Unix y a la vez indicar que GNU es algo diferente. Técnicamente GNU es como Unix. Pero GNU le da a sus usuarios libertad, y Unix no.

Llevó muchos años de trabajo por parte de cientos de programadores, para desarrollar este sistema operativo. Algunos fueron contratados por la Free Software Foundation y por las compañías libres de software libre; la mayoría sin embargo eran voluntarios. Algunos se han vuelto famosos mientras que la mayoría son conocidos principalmente entre sus pares, conocidos por otros hackers que usan o trabajan sobre su código. Todos juntos han ayudado a liberar el potencial de uso de las computadoras para toda la humanidad.

En 1991 fue desarrollado el último componente esencial de un sistema del estilo de Unix: Linux, el kernel libre escrito por Linus Torvalds. Hoy, la combinación de GNU y Linux es usada por millones de personas alrededor del mundo y su popularidad está creciendo. El escritorio gráfico de GNU hace ahora del sistema GNU/Linux un sistema casi tan fácil de usar como cualquier otro sistema operativo.

Pero nuestra libertad no está asegurada para siempre. El mundo no se queda quieto y no podemos contar con el hecho de tener libertad dentro de cinco años solamente porque la tenemos hoy. El software libre enfrenta desafíos difíciles y peligros. Serán necesarios ciertos esfuerzos para preservar nuestra libertad, así como llevó esfuerzo obtener esta libertad al principio. Mientras tanto el sistema operativo es solamente el comienzo, ahora necesitamos aplicaciones libres que manejen toda la gama de trabajos que los usuarios quieren hacer. No importa el trabajo que uno quiera hacer con la computadora no se debería renunciar a la libertad de hacerlo.

UNESCO siempre ha apoyado la extensión y la diseminación del conocimiento humano. Le agradezco a UNESCO por reconocer que, en el dominio del software, el software libre disemina el conocimiento humano de una manera imposible de lograr por el software que no es libre

Richard Stallman

CONTENIDO

PRÓLOGO

A partir de las recomendaciones surgidas del LACFREE 2005 (Conferencia Latinoamericana y del Caribe sobre Desarrollo y Uso del Software Libre), que tuviera lugar en Recife, Brasil, la UNESCO percibió la necesidad de proveer información concreta sobre las experiencias en el uso de software libre.

Tanto los telecentros como los centros comunitarios multimedia están interesados en introducir más soluciones de software libre para sus usuarios en las comunidades locales.

Los docentes buscan aplicaciones de código abierto que les permitan incluir formas de aprendizaje interactivo en los temas de la curricula. Modelos, simulaciones, gráficos, multimedia, comunicación y diseño web requieren herramientas de software libre accesibles al mundo educativo. Los sistemas de gerenciamiento de aprendizaje de software libre poseen una amplia funcionalidad profesional y facilitan la organización institucional del aprendizaje virtual a nivel de escuelas y universidades.

Las prioridades de la Información para Todos, apuntan a la reducción de la brecha digital y, en muchos países, son parte de la política nacional para el diseño de sociedades del conocimiento. El software libre puede contribuir con soluciones a esas demandas de inclusión digital y social, por ejemplo ofreciendo herramientas para que las personas con déficit visual o auditivo puedan aprovechar las bibliotecas y escuelas públicas. Al mismo tiempo los expertos ven al software libre como una herramienta para desarrollar capacidades competitivas en ingeniería de software y en las industrias vinculadas a nivel nacional.

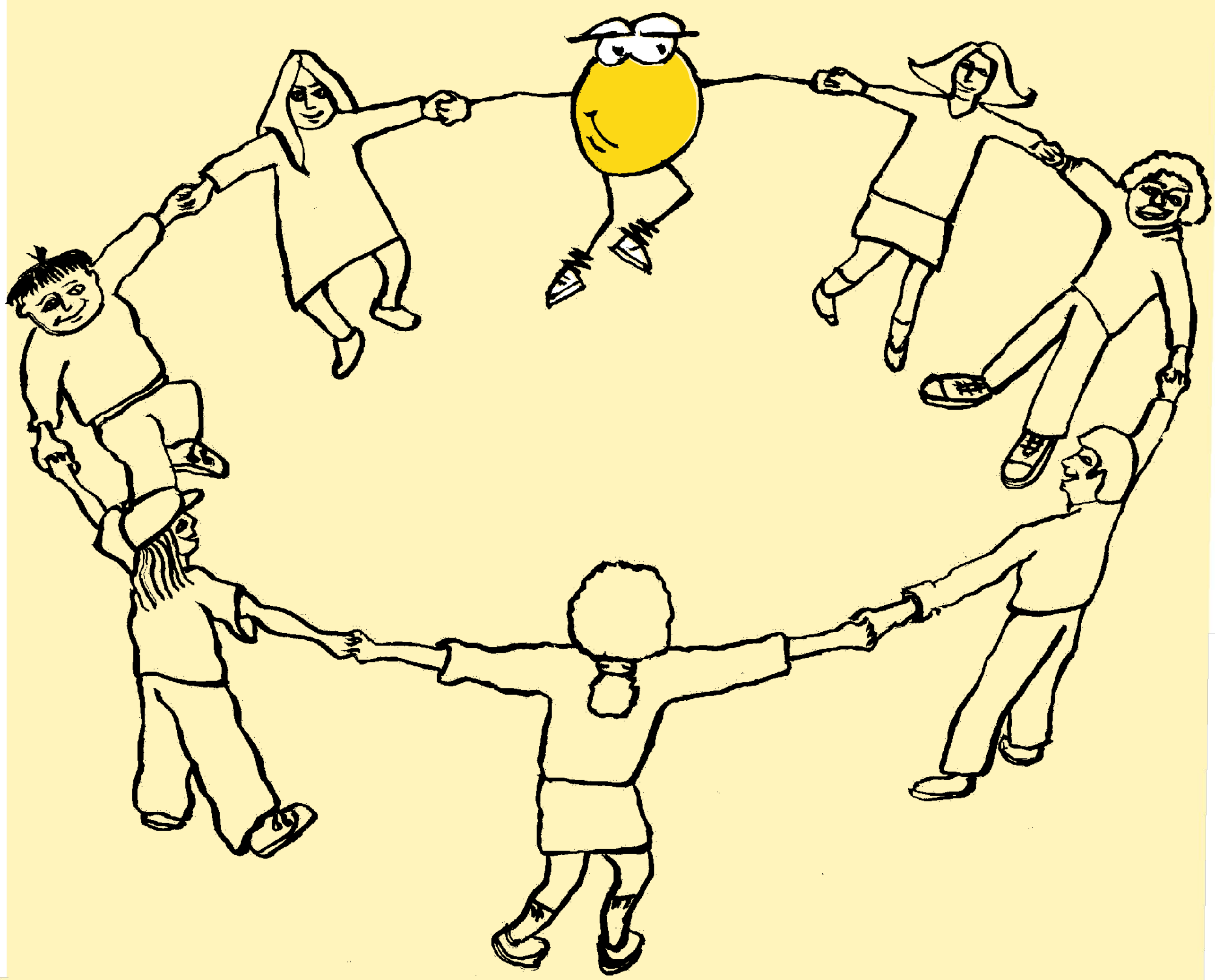
Esta guía práctica presenta definiciones de lo que es el software libre de acuerdo a la Free Software Foundation y explica los procesos de desarrollo del software como una actividad basada en la comunidad.

Las recomendaciones de carácter práctico para migrar hacia el software libre facilitan el realizar estudios de factibilidad. La guía desarrolla cinco argumentos en cómo las administraciones públicas y el concepto de gobierno electrónico pueden beneficiarse del uso del software libre y los formatos abiertos.

Finalmente, esta guía para América Latina y Caribe nos ayudará a entender el papel fundamental que el Software Libre pueden jugar en el uso de las tecnologías de la comunicación y la información para un desarrollo sostenible.

Günther Cyranek
Consejero de Comunicación e Información en el MERCOSUR y Chile
Oficina Regional de Ciencia de la UNESCO para América Latina y Caribe
UNESCO - Montevideo
Agosto 2007

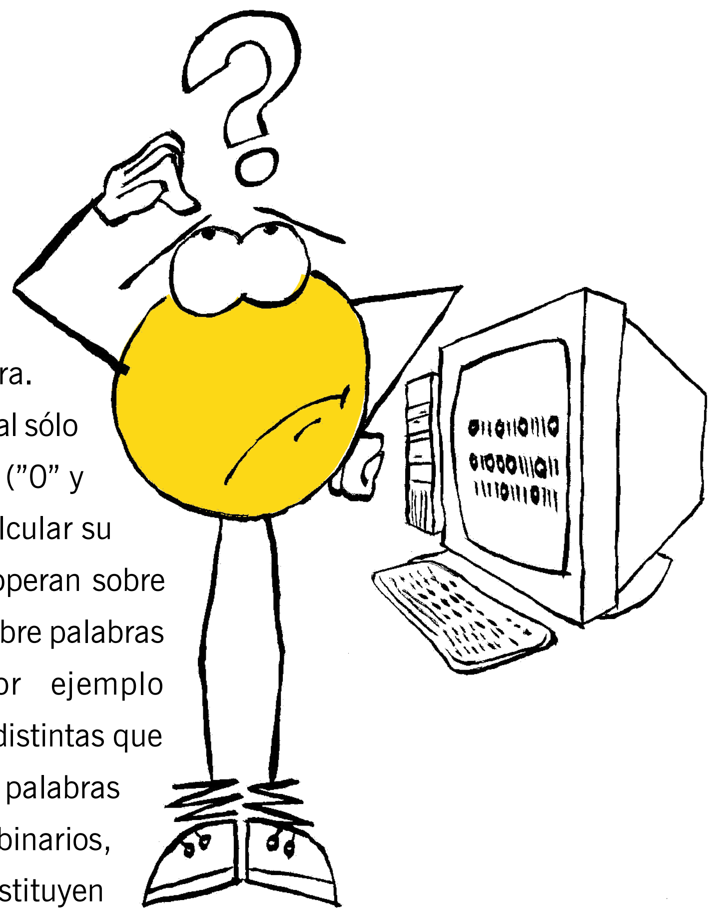
INTRODUCCION



UNA BREVE INTRODUCCION

Es muy difícil comprender en forma cabal la importancia y proyección del Software Libre sin antes entender qué es el software, cómo se desarrolla y su rol dentro de los sistemas de procesamiento de datos. Para esto necesitamos explorar conceptos básicos vinculados a la informática, tales como tecnología, programa, código fuente y código objeto.

LA COMPUTADORA



Por mucho que se hable de "dispositivos inteligentes", pocas cosas hay más tontas que el procesador de una computadora. Si vamos a su esencia, un procesador elemental sólo sabe manejarse con un alfabeto de dos letras ("0" y "1") y dadas dos "letras" de este alfabeto, calcular su suma. Por lo general, las computadoras no operan sobre "letras" individuales (llamadas "bits"), sino sobre palabras de ocho letras, llamadas "bytes", por ejemplo "01100101". Hay un total de 256 palabras distintas que se pueden escribir con ocho bits. Estas 256 palabras que, si las interpretamos como números binarios, representan los números del cero al 255, constituyen el vocabulario completo de la computadora.

¿Cómo es posible que con sólo 256 palabras en su vocabulario una computadora pueda hacer tantas cosas? ¿Cómo es posible que sólo sumando pueda hacer cálculos complejos?



El secreto está en que la computadora es programable. La computadora sabe obedecer órdenes, y tiene la capacidad de ejecutar millones de instrucciones en un solo segundo, ejecutando operaciones sobre los millones de bytes que constituyen su memoria. Combinando instrucciones, podemos escribir programas que combinen e interpreten las palabras almacenadas en la memoria de distinta manera de acuerdo al contexto: como cada una de las letras de este texto, como números, o notas musicales, o colores. Enviando los datos de la manera adecuada a una pantalla, una impresora o una tarjeta de sonido, el programa puede impresionar nuestros sentidos. Aceptando datos desde el teclado y el ratón, puede responder a nuestros comandos.

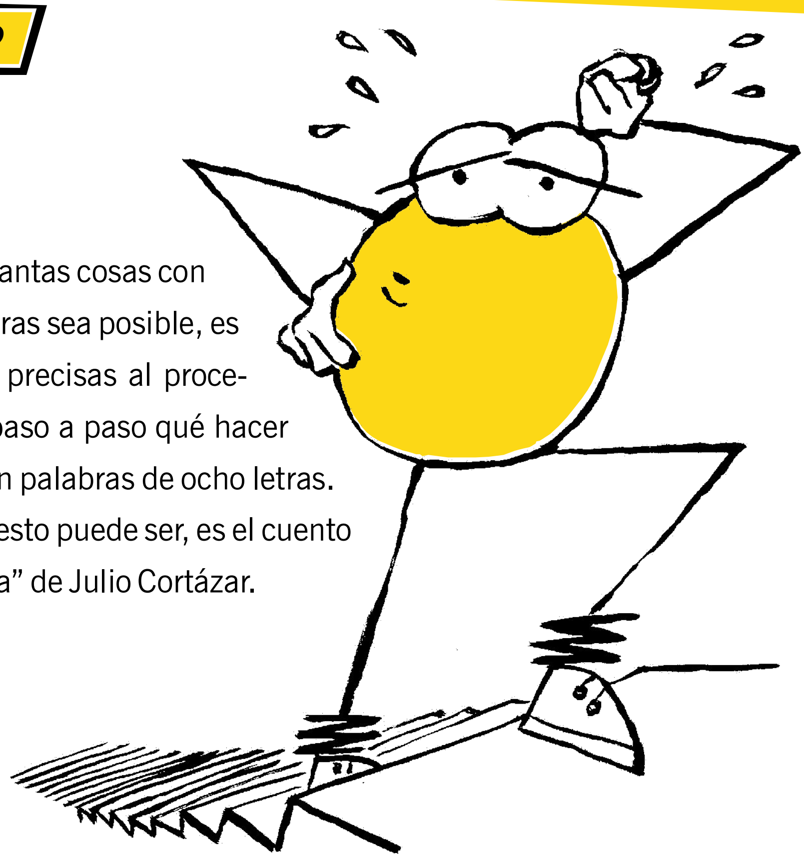
Las computadoras son dispositivos de propósito general, a las que podemos configurar para cumplir tareas específicas: imprimir documentos, tocar música, mostrar videos, interconectar redes, por sí mismas son incapaces de realizar ninguna. Para ello, necesitan que alguien les provea instrucciones detalladas acerca de cómo interpretar la información y cómo comunicarse con el usuario. Un programa es un conjunto de instrucciones para la computadora, que le permite llevar adelante una tarea específica. Ese programa se escribe, como se escribe este texto o cualquier relato. Los encargados de escribir esos textos son los programadores, que dominan un determinado lenguaje de programación para realizar esta tarea, como nosotros dominamos el español, su gramática y su sintaxis para construir este texto.



¿QUÉ ES EL CÓDIGO FUENTE?

EL PROGRAMA

Para que el "milagro" de hacer tantas cosas con un lenguaje de sólo 256 palabras sea posible, es necesario darle instrucciones precisas al procesador de la computadora, indicarle paso a paso qué hacer con esos unos y ceros, combinados en palabras de ocho letras. Un buen ejemplo de lo complejo que esto puede ser, es el cuento "Instrucciones para subir una escalera" de Julio Cortázar.

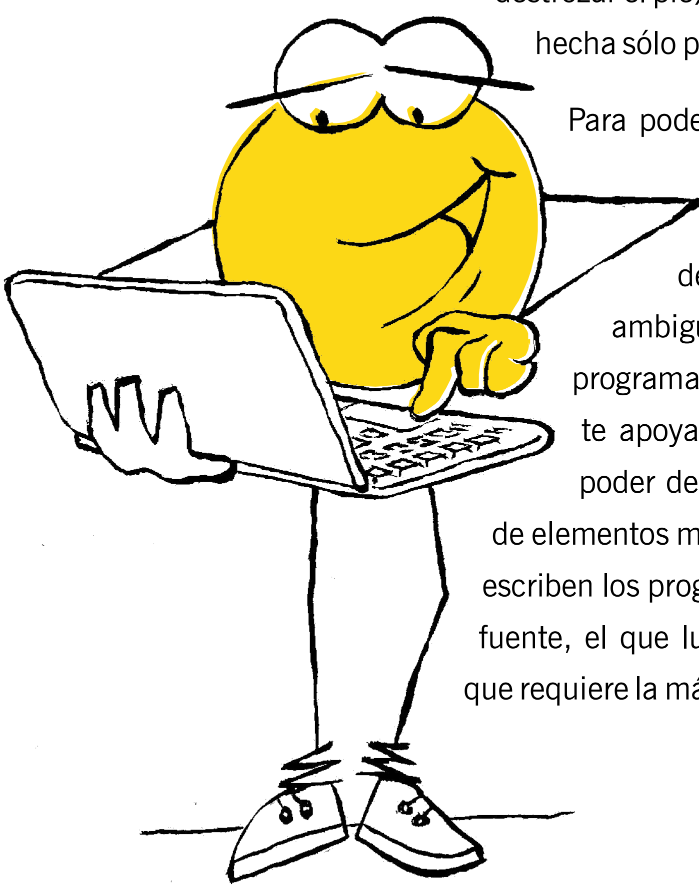


"Para subir una escalera se comienza por levantar esa parte del cuerpo situada a la derecha abajo, envuelta casi siempre en cuero o gamuza, y que salvo excepciones cabe exactamente en el escalón. Puesta en el primer peldaño dicha parte, que para abreviar llamaremos pie, se recoge la parte equivalente de la izquierda (también llamada pie, pero que no ha de confundirse con el pie antes citado), y llevándola a la altura del pie, se le hace seguir hasta colocarla en el segundo peldaño…"?

Julio Cortázar

En realidad, por detalladas que suenen las instrucciones de Julio Cortázar, una computadora que intentara seguirlas las encontraría exasperantemente abstractas. Aún si dejáramos de lado el detalle de que "levantar" no tiene sentido sin antes ponernos de acuerdo dónde es "arriba", y suponiendo que sabemos qué es un "pie", una simple instrucción como "levantar el pie" deja a la computadora fuera de combate. Hay que levantarlo, sí, ¿pero cómo? Más acertado sería enumerar las articulaciones de la pierna que deben moverse, y especificar a qué ángulo deben flexionarse en cada instante del tiempo, pero esto tampoco sería suficientemente detallado. Lo que en realidad hay que describir es el detalle de los impulsos nerviosos que hay que enviarle a cuáles músculos... con menos de esto, la máquina no sabría por dónde empezar.

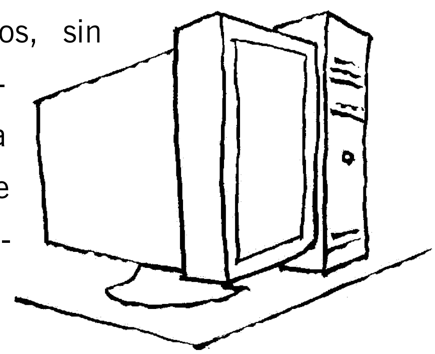
Eso sí: provista con instrucciones suficientemente detalladas, la computadora las cumple rápidamente y sin cuestionarlas. Si las instrucciones contienen alguna equivocación que la llevan a patear la escalera hasta convertirla en astillas (o, si es una escalera dura, hasta destrozar el pie), lo hará sin dudar, pues la computadora está hecha sólo para obedecer órdenes.

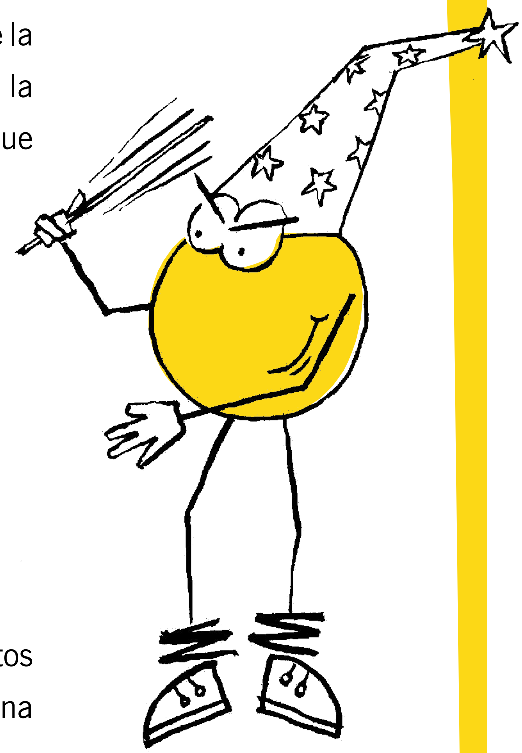


Para poder manejar la complejidad que requiere dar una descripción tan detallada y precisa, la exquisita prosa de Julio Cortázar no sólo es demasiado abstracta, sino peligrosamente ambigua. En vez de usar castellano, o inglés, los programadores utilizan lenguajes formales fuertemente apoyados en la matemática, y derivan de ella el poder de expresar relaciones abstractas en términos de elementos más concretos. En estos lenguajes formales se escriben los programas y se desarrolla el denominado código fuente, el que luego se traduce al lenguaje super detallado que requiere la máquina, el lenguaje máquina.

LENGUAJE DE MÁQUINA

A diferencia de los objetos materiales, que se *fabrican*, los programas se *escriben*. Quizás parte de la fascinación que la programación ejerce sobre quienes la practican se deba a la magia aparente de lograr que una máquina cumpla las órdenes que le impartimos verbalmente. Para que pueda entender lo que le ordenamos, sin embargo, debemos comunicarnos con ella de una manera especial: el procesador, dentro de cada computadora, sólo comprende instrucciones escritas en su propio lenguaje, denominado lenguaje de máquina (a veces mencionado como código binario o código objeto). Existen muchos tipos distintos de procesador y cada uno de ellos tiene un lenguaje de máquina distinto, diseñado por su fabricante. Esa es la razón por la que no es posible ejecutar cualquier programa en cualquier computadora.





Los programas escritos en lenguajes de máquina son, básicamente, larguísimas listas de números. Cada uno de estos números representa una operación elemental (por ejemplo, 1: sumar, 2: restar, 3: almacenar en memoria, etc), o un operando, o una combinación de ambas cosas. Para que el lector pueda darse una idea concreta de cómo es un programa codificado en lenguaje de máquina, lo ilustraré con un ejemplo: lo que sigue es una pequeña porción de un programa, escrito en el lenguaje de máquina de los procesadores de la familia Intel funcionando bajo el sistema operativo GNU/Linux. Este programa imprime la raíz cuadrada de un número cualquiera. Si el

número resulta ser negativo, el programa imprime un mensaje de error (recordemos que no es posible sacar la raíz cuadrada de un número negativo):

2212858197	1171855596	3673086216	2665537513
250282615	1680082119	3892839557	4294967036
1171856363	605871368	4294901736	610065919
604292868	134514050	4294893544	1438894591

Una computadora moderna está en condiciones de ejecutar millones de instrucciones de máquina como estas por segundo. Construir un programa complejo a partir de instrucciones tan elementales y codificadas en un formato tan poco amigable es, sin embargo, una tarea muy complicada para un ser humano. También es extremadamente difícil entender cómo funciona un programa si sólo disponemos de él en lenguaje de máquina. Para comprender cómo funciona el programa citado más arriba no sólo es necesario recordar qué número corresponde a cual operación, también debemos identificar cuáles partes de los números representan operaciones, cuáles representan operandos, cuáles datos, y muchas otras cosas más. Debido a detalles técnicos, modificarlo es mucho más difícil aún, ya que para eliminar o agregar instrucciones no basta con insertarlas o borrarlas en la lista: también hay que modificar los valores de muchas de las demás instrucciones. Por ello, modificar un programa del que sólo tenemos una copia en lenguaje de máquina, o traducirlo para que funcione en una máquina con un lenguaje de máquina distinto, es una tarea virtualmente impracticable.

Dado que las personas tienen serias dificultades para leer, escribir y modificar programas en lenguaje de máquina, pero son esas mismas personas las que deben escribir y mantener los programas, se inventaron los lenguajes de programación. A diferencia de los lenguajes máquina, que están diseñados con el único objetivo de facilitar al procesador la ejecución de las instrucciones a gran velocidad (podríamos llamarlo también lenguaje de ejecución), la finalidad de los lenguajes de programación es facilitar a los seres humanos la comprensión, la escritura y la modificación de los programas. El objetivo primordial de un programa expresado en un lenguaje de computación no es ser ejecutado directamente por una computadora (lo que es imposible), sino comunicar, de manera comprensible para un ser humano, lo que una computadora debe hacer para resolver un determinado problema.

Dado que la naturaleza de la computación está íntimamente ligada a las matemáticas, los lenguajes de programación generalmente son una mezcla un tanto idiosincrática de notación matemática entrelazada con una gramática rudimentaria, por lo general basada (remotamente) en el inglés. A modo de ejemplo, veamos cómo se expresa el mismo programa citado más arriba en un lenguaje de programación muy difundido, llamado “C”:

```
/* Esta función imprime la raíz cuadrada de su argumento */
static void printsqrt(float x) {
  if (x < 0) /* la raíz de un numero negativo es imaginaria */
    printf("El numero es menor que cero!\n");
  else    /* el numero es positivo, todo bien */
    printf("%f\n", sqrt(x)); }
```

Aún para personas que no conocen el lenguaje C, este texto es más comprensible que el programa en lenguaje de máquina de más arriba, ya que hay algunos elementos claramente distinguibles.

Quizás lo más notable para el lego sean las notas aclaratorias en castellano, entre los símbolos “/*” y “*/”. Estos textos (llamados comentarios) no afectan la función del programa, pero sí son importantes para dejar plasmada la intención del programador. En otras palabras, cumplen el rol de “notas adhesivas”, en las que el programador aclara el objetivo de cada parte del programa. Esto es muy útil cuando otro programador intenta comprender cómo funciona el programa, o incluso cuando el mismo programador debe revisar ese mismo programa un tiempo después de haberlo escrito.

Más allá de los comentarios, si sabemos que en inglés “*if*” quiere decir “si” (condicional, no afirmativo), y que “*else*” quiere decir “de lo contrario”, es relativamente sencillo advertir que la sentencia “*if (x<0)*” determina si el valor de la variable “x” es menor que cero y que, dependiendo del resultado, el procesador ejecutará la sentencia que está entre el “*if*” y el “*else*”, o de lo contrario la que sigue al “*else*”. También es fácil identificar los mensajes que serán mostrados al usuario, aunque el segundo de ellos (“%f\n”) contenga en realidad una codificación propia e idiosincrática del lenguaje C: son los que están entre comillas. También es plausible, para quienes hablan inglés, que el nombre de la función sqrt() es una abreviatura de *square root*, o “raíz cuadrada”.

El mismo programa puede escribirse en distintos lenguajes de programación.

Por ejemplo, en el lenguaje Python :

```
def printsqrt(x=0):
   """Esta función imprime la raíz cuadrada de su argumento"""
   if (x < 0): # la raíz de un numero negativo es imaginaria
      print "El numero es menor que cero!"
   else:               # el numero es positivo, todo bien
      print math.sqrt(x)
```

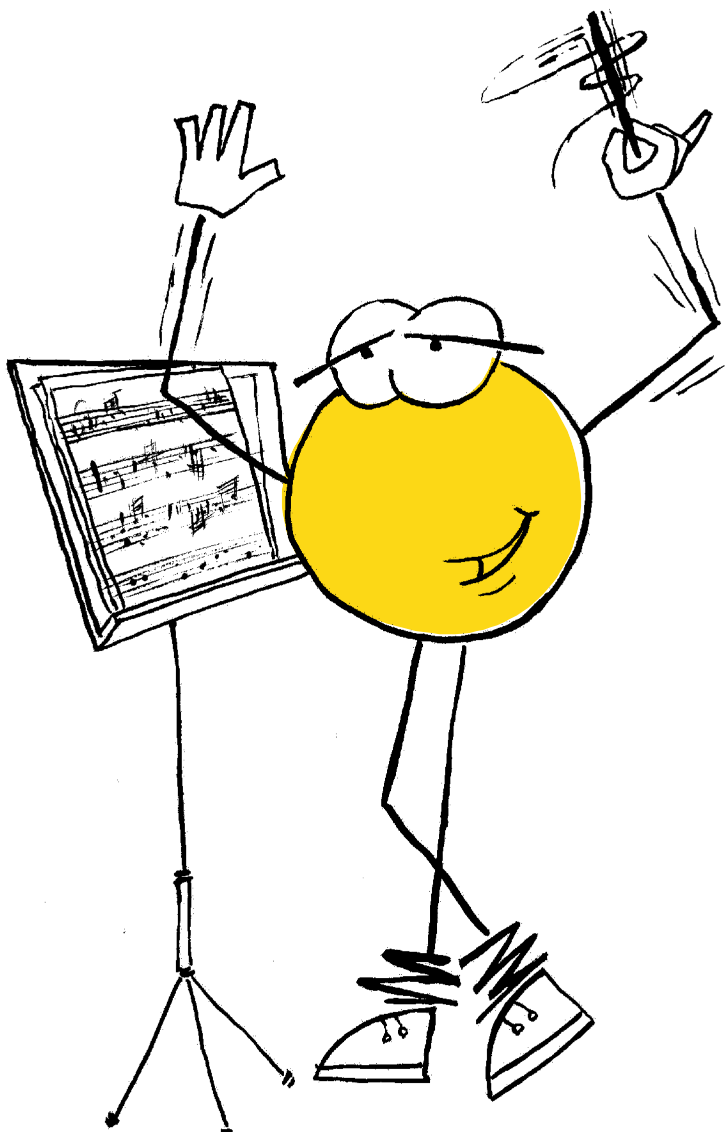
O en el lenguaje Smalltalk:

```
“Esta función imprime la raíz cuadrada de su argumento"
printsqrt: x
   (x < 0)
      ifTrue: "la raíz de un numero negativo es imaginaria"
         [Transcript show:'El numero es menor que cero!']
   ifFalse: "el numero es positivo, todo bien"
         [Transcript show: x sqrt]
```

De un lenguaje de programación a otro, las convenciones cambian, y también algunos aspectos técnicos, pero en todos ellos reconocemos elementos comunes, y sobre todo vemos que su objetivo es facilitar la confección, comprensión y modificación del programa, al permitir que el programador trabaje en un nivel de abstracción que es confortable a una mente humana.

Leyendo el código fuente de un programa es donde resulta más fácil apreciar la naturaleza cultural del software, tal como lo reconoce la Carta de Recife, firmada por los asistentes a la II Conferencia Latinoamericana y del Caribe de Software Libre (LACFREE II). El programa no resuelve un problema, describe una solución, de la

misma manera que una ecuación describe una relación entre sus términos, o que una partitura musical describe una sinfonía. El programa en sí no es activo, no hace nada: es la máquina la que, siguiendo el mecanismo descrito por el programa, actúa sobre el medio para resolver el problema.



COMPILACIÓN



Los programas escritos en un lenguaje de programación no son comprensibles directamente por una computadora. Recordemos que éstas sólo saben obedecer instrucciones codificadas en su lenguaje de máquina. Antes de poder ejecutar un programa escrito en un lenguaje de programación, debemos traducirlo al lenguaje de la máquina sobre la que queremos que corra. Para cada combinación de procesador, lenguaje y sistema operativo existen traductores automáticos, llamados compiladores. Se trata de programas que leen un programa escrito en un lenguaje de programación y, a partir de él, generan uno escrito en el lenguaje de ejecución adecuado para una determinada combinación de procesador y sistema operativo. El programa en lenguaje de máquina que vimos más arriba, de hecho, es el resultado de pasar el texto del programa C que vimos a continuación por un compilador llamado "**gcc**".

Aquí se vuelve evidente otra gran ventaja de los lenguajes de programación sobre el lenguaje de máquina: si mi programa está expresado en un lenguaje de máquina, sólo podré ejecutarlo en máquinas equipadas con un determinado tipo de procesador. Por el contrario, si lo escribí en un lenguaje de programación, en principio basta con hacerlo traducir por el compilador adecuado para que el programa pueda correr sobre el procesador que yo quiera.

"CÓDIGO FUENTE"

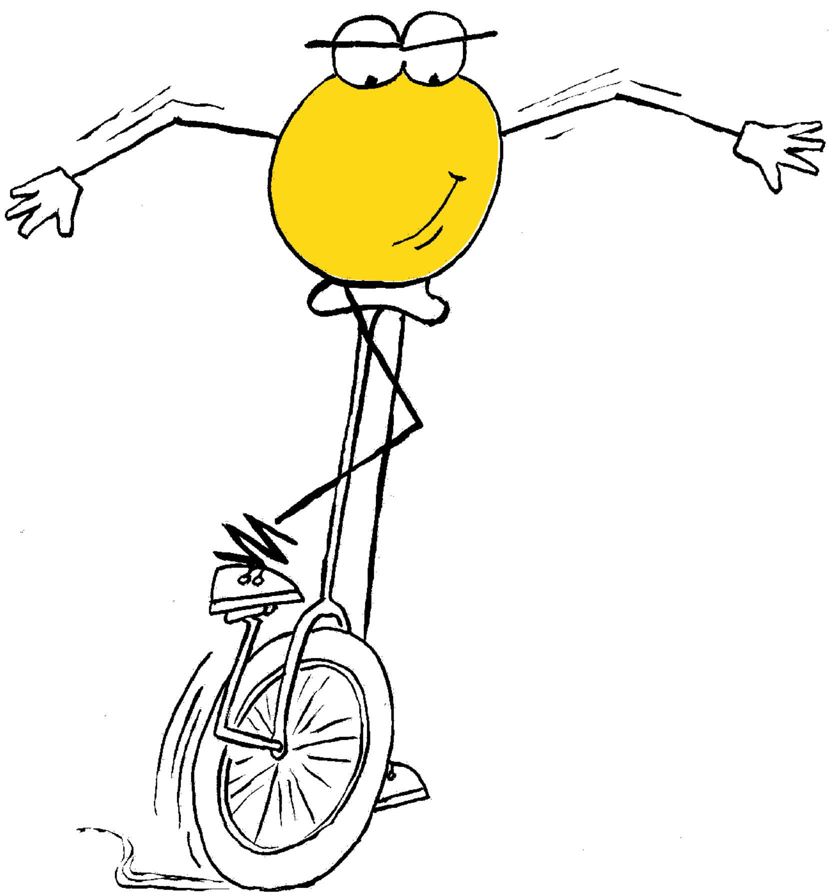
En inglés, se conoce al programa escrito en lenguaje de programación como *source code*, y al programa expresado en lenguaje de máquina como *object code* (código objeto) o *executable code* (código ejecutable). En castellano, a menudo se traduce *source code* con la frase "código fuente", que no es completamente fiel a la intención de la expresión inglesa. Tendría más precisión técnica traducirlo como "texto original" del programa, ya que se trata del texto tal como lo escribió el programador, mientras que el código ejecutable es el fruto de una traducción automática realizada por un compilador.

Si bien es posible ejecutar un programa en la computadora adecuada contando sólo con el código ejecutable, cuando se trata de comprender el funcionamiento de un programa, de modificarlo, o de hacerlo funcionar en una máquina diferente, es imprescindible disponer de su texto original, es decir de su "código fuente".



CÓDIGO FUENTE Y SOFTWARE LIBRE

Para ser considerado libre, un programa debe ser distribuido de tal modo que el usuario pueda, entre otras cosas, estudiar el modo de funcionamiento del programa, adaptarlo a sus necesidades y distribuir, bajo las mismas condiciones, programas derivados. Para que estas libertades sean practicables, no basta con que la licencia del programa las permita. Además, es necesario que el código fuente del programa esté a disposición del usuario, ya que de lo contrario las tareas de comprender, adaptar y mejorar el programa se vuelven tan complicadas que es casi lo mismo que si estuvieran prohibidas. Por eso la definición de Software Libre elaborada por la Free Software Foundation aclara que un programa no puede ser considerado libre si su código fuente, su texto original, no está disponible.

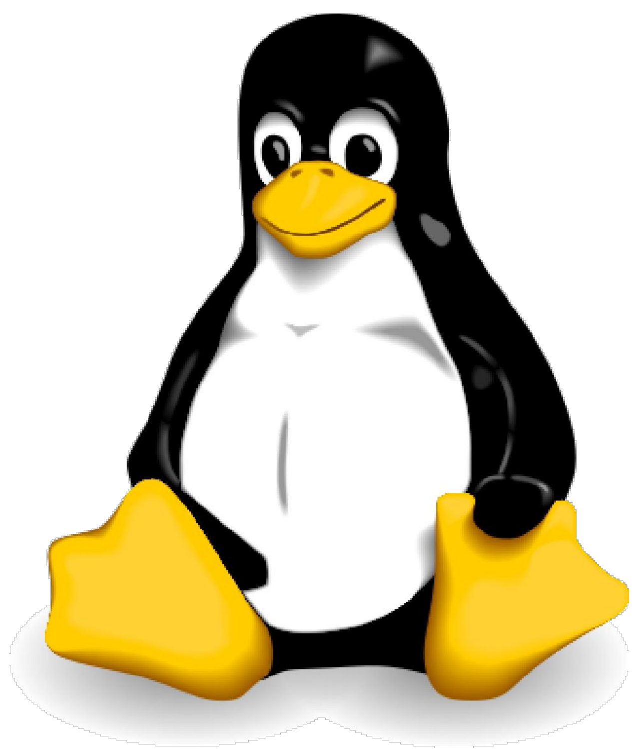


SOFTWARE LIBRE









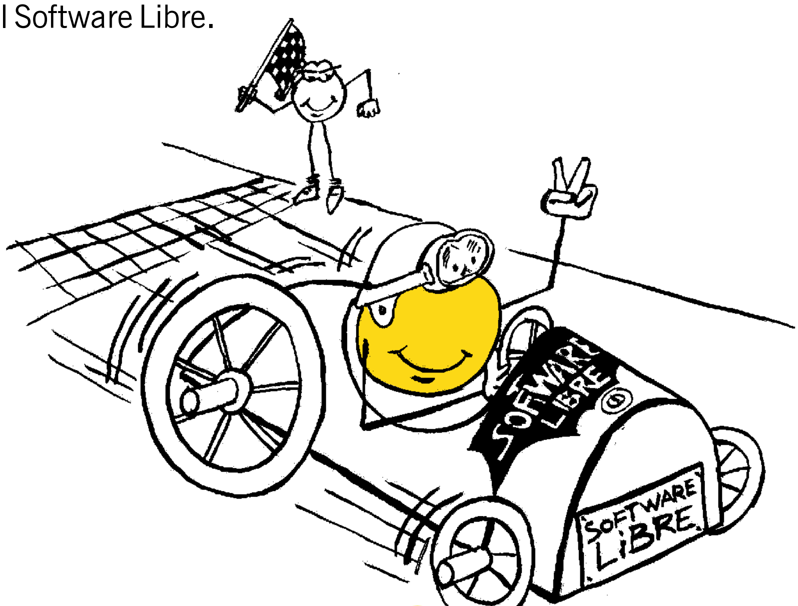
SOFTWARE LIBRE

¿UN SOFWARE DISTINTO?

Es común identificar el Software Libre con una determinada tecnología, por lo que se suele sostener que el Software Libre es Linux y, por lo tanto, los impulsores del Software Libre están a favor de Linux y en contra de otros sistemas operativos. Contrariamente a esta idea, es importante aclarar que Software Libre no es una determinada tecnología, no es un tipo de programa de computadora ni un sistema operativo.

Si Usted se sienta frente a una computadora que ejecuta dos programas y le preguntamos cuál de ellos es Software Libre y cuál no lo es, no le será posible dar una respuesta sin leer el documento de licencia de cada uno de los programas. Así, por ejemplo, una suite de oficina libre y una no libre hacen prácticamente lo mismo. Igual pasa con un navegador web libre o uno no libre. Las posibles diferencias funcionales – cuando las hay – no son el dato relevante para identificar si un programa es libre o no.

El Software Libre se define por su tipo de licenciamiento. Por lo que podemos entonces llamar “software licenciado bajo condiciones libres” al Software Libre.



¿UNA FORMA DISTINTA DE DISTRIBUIR SOFTWARE?

Entonces, para poder distinguir un programa licenciado bajo condiciones libres de uno licenciado con una licencia no libre, es obvio, debemos recurrir al documento de licencia y saber distinguir cuáles son los permisos y las condiciones de los mismos que transforman un programa de computadora en Software Libre.

No será raro entonces asistir a una reunión de partidarios del Software Libre y encontrarse en una discusión o presentación sobre condiciones de licencias de software. No es raro, tampoco, ver a los partidarios del Software Libre poniendo un celo exacerbado en reconocer los derechos de los autores de los programas, en respetar los Derechos de Autor y alzar su dedo contra quien no lo hace o se toma a la ligera estos temas.

Software libre

Simplificando al máximo, podemos decir que Software Libre es un software o programa de computación cuya licencia nos permite ejercer una serie de libertades:





- La libertad de ejecutar el programa con cualquier propósito.
- La libertad de estudiar cómo funciona el programa y adaptarlo a las necesidades propias(para lo cual es una precondición el acceso al código fuente).
- La libertad de redistribuir copias del programa y de ese modo ayudar a otros.
- La libertad de mejorar el programa y liberar esas mejoras al público beneficiando así a toda la comunidad (para lo cual es una precondición el acceso al código fuente).

Y sólo exige una cosa, en el caso de la licencia GPL: que si distribuimos el programa resultante de una modificación, éste se distribuya bajo las mismas condiciones del programa original. Las licencias que contienen esta condición son llamadas “licencias Copyleft”, y su objetivo es evitar que se distribuyan obras derivadas bajo licencias privativas. Como contraparte podemos decir que software propietario o privativo (dado que nos priva de libertades) es el que nos priva de alguna de las libertades antes expuestas.

¿CÓMO SURGE EL MOVIMIENTO DEL SOFTWARE LIBRE?

El origen tiene un nombre: Richard Stallman, un programador del MIT (Instituto de Tecnología de Massachusetts). Él se había acostumbrado – como muchos en su comunidad de hackers - a colaborar con otros desarrolladores intercambiando código fuente. De esa manera, cooperando con otros desarrolladores, lograba buenos resultados. Esto último era una práctica muy común en los inicios de los 70, cuando Stallman comenzó a trabajar en el MIT, pero las cosas fueron cambiando, en buena medida debido a que el desarrollo de software se convirtió en un gran negocio. Al llegar a los 80, el código fuente había pasado de ser un elemento de intercambio entre los programadores a ser considerado un secreto estratégico para varias empresas.

Es en ese contexto que Richard Stallman publica, en 1983, en el grupo de noticias net.unix-wizards, el primer llamado al desarrollo del proyecto GNU, en el que decía entre otras cosas:

"Iniciando este día de Acción de Gracias voy a escribir un sistema de software completo compatible con Unix llamado GNU (siglas de Gnu No es Unix), y lo distribuiré libre a quienes puedan usarlo. Considero que la regla de oro exige que si yo quiero un programa debo compartirlo con otras personas que también lo quieren. No puedo, conscientemente, firmar un acuerdo de confidencialidad o un acuerdo de licencia de software. Para que yo pueda continuar utilizando las computadoras sin violar mis principios, he decidido reunir suficiente Software Libre de manera de poder continuar sin necesidad de utilizar algún software que no sea libre.(…) estoy buscando personas para quienes el ayudar a la humanidad sea tan importante como el dinero."



Y al parecer las encontró, veinte años después el movimiento GNU tiene una difusión universal. Lo que Stallman deseaba era: “retornar al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras”.

Los programas de computadora, como obra intelectual, están bajo el mismo régimen de derecho de autor tal como un libro o una canción. Así, las mismas restricciones que se aplican a un libro son aplicables al software: está prohibida su reproducción y distribución sin permiso del autor. Como para instalar un programa en una computadora es necesario copiarlo, el software no se puede instalar sin permiso del autor.

Los autores pueden otorgar permisos a su público, sus usuarios, mediante el uso de licencias. Algunos autores utilizan licencias para otorgar el permiso de instalar el programa, pero condicionan el permiso a una serie de exigencias: el usuario no puede modificar el programa, ni averiguar cómo funciona, por ejemplo. Otros autores, en cambio, aprovechan las licencias para otorgar a los usuarios amplias facultades sobre el software, y se cuidan de no restringirlos adicionalmente. Algunos permiten la reproducción ilimitada, otros prestan atención a no limitar el derecho del usuario a estudiar el programa.

La única manera de determinar si un programa es libre o no, entonces, es revisando la licencia que lo acompaña.



Richard M. Stallman

Es el padre fundador del movimiento e ideólogo principal de la filosofía del Software Libre. Nació en los EEUU en 1953, estudió Física en Harvard y se integró al Laboratorio de Inteligencia Artificial del MIT, puesto del cual dimitió a mediados de los años 80 para dedicar todo su esfuerzo a la construcción de Software Libre.

En 1984 inició el proyecto GNU (GNU's not Unix) con el que dio el primer puntapié para la construcción de Software Libre. Fundó y preside la Free Software Foundation, principal organización que vela por las políticas y la filosofía del movimiento y que promueve el uso de licencias de Software Libre. Además de sus méritos como programador de piezas clásicas como el compilador GCC o el editor de textos Emacs, RMS ha hecho enormes contribuciones en el campo jurídico, filosófico y político, entre ellas, la concepción del concepto de "Copyleft" y la construcción de licencias de software y documentación como la GNU/GPL (usada para distribuir el núcleo Linux entre otros) y la GNU/GFDL (usada por ejemplo en Wikipedia).
Más información en http://es.wikipedia.org/wiki/Stallman

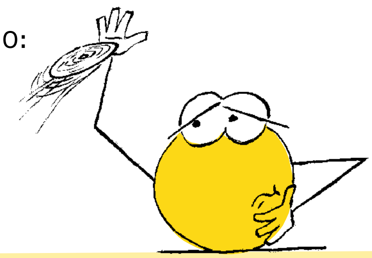
LICENCIAS MUY DISTINTAS

Existen muchas licencias distintas dentro del ámbito del Software Libre, la más utilizada y conocida es la GNU GPL desarrollada por la FSF (Free Software Foundation). Basada en las libertades antes expuestas, dicha licencia incluye la protección para que el programa no pueda ser convertido en software privativo, al plantear expresamente que:

> "Cada vez que usted redistribuya el Programa (o cualquier trabajo derivado del Programa), el receptor automáticamente recibe la licencia por parte del licenciatario original para copiar, distribuir o modificar el Programa sujeto a estos términos y condiciones. Usted no puede imponer ninguna otra restricción a los receptores limitando los derechos garantizados en esta Licencia."



Una forma clásica de expresar en un programa que se encuentra bajo la licencia GNU Licencia Pública General (GPL) lo podemos ver con el siguiente ejemplo:



```
Sumar - Programa para sumar dos números
Copyright (C) 2007  Nombre del AutorEste programa es
Software Libre; Usted puede redistribuirlo y/o
modificarlo bajo los términos de la GNU Licencia
Pública General (GPL) tal y como ha sido públicada
por la Free Software Foundation; o bien la versión 2
de la Licencia, o (a su opción) cualquier versión
posterior.
Este programa se distribuye con la esperanza de que
sea útil, pero SIN NINGUNA GARANTÍA; tampoco las
implícitas garantías de MERCANTILIDAD o ADECUACIÓN A
UN PROPÓSITO PARTICULAR. Consulte la GNU General
Public License (GPL) para más detalles. Usted debe
recibir una copia de la GNU General Public License
(GPL)junto con este programa; si no, escriba a la
Free Software Foundation Inc.
51 Franklin Street, 5° Piso, Boston, MA 02110-1301,
USA. Datos del autor.
```

Existen otras licencias dentro del ámbito del Software Libre, licencias que también brindan las libertades antes expuestas, pero a diferencia de la GNU(GPL) no impiden que se pueda utilizar el trabajo de programación realizado e incluirlo en un nuevo software que no brinde las libertades antes expuestas. Un ejemplo es la licencia BSD (Berkeley Software Distribution) desarrollada por la Universidad de Berkeley y que permite el uso del código desarrollado en un software derivado que sea licenciado como privativo.

¿CÓMO SELECCIONAR SOFTWARE LIBRE?




SL

¿CÓMO SELECCIONAR SOFTWARE LIBRE?

Tal vez uno de los mayores desafíos que presenta el Software Libre es la variedad de aplicaciones licenciadas para cada una de las ramas de la actividad humana.

Se calcula que actualmente existen alrededor de cien mil proyectos de desarrollo de Software Libre, algunos se encuentran en etapa de planificación y otros ya se encuentran maduros dando frutos y pueden ser utilizados para cubrir una amplia gama de necesidades.

Sería muy pretencioso para esta guía recomendar Software Libre específico que cubra toda la variedad de realidades que pueden requerir soluciones particulares. De hecho, le recomendamos que, si Usted está a cargo de una organización que plantea usar software licenciado como libre en forma exclusiva, se vincule con una comunidad de usuarios que pueda brindarle contención y asesoramiento.

No obstante, haremos algunas recomendaciones sobre como seleccionar Software Libre y le haremos algunas recomendaciones de software que ha sido probado y utilizado por muchas otras organizaciones con buenos resultados.

El Software Libre puede ser seleccionado siguiendo algunos criterios que le permitirán conocer si está frente a una aplicación que puede ser útil y cubrir sus necesidades. Dichos criterios han sido los utilizados por nosotros para recomendar el software en esta guía:

Proyecto vivo (en actividad). En este sentido es importante saber si existe una importante comunidad de desarrolladores apoyando el proyecto en el momento de su elección. En el caso del software privativo, es común encontrar que lo único que existe detrás del producto es el departamento de desarrollo de software de una empresa, con lo cual el riesgo de pérdida de soporte y desarrollo futuro es grande. De hecho, suelen llegar noticias regularmente de programas discontinuados por su propio desarrollador. En Software Libre esto se previene evaluando la vitalidad de una comunidad alrededor de cada proyecto.

Proyecto más aceptado (incluido por distribuciones de Software Libre). Lo cual es un índice de la evaluación desarrollada por otros integrantes de la comunidad del Software Libre o empresas que crean distribuciones.

Proyecto estable y maduro (muchas veces no están vivos, pero son estables y confiables).

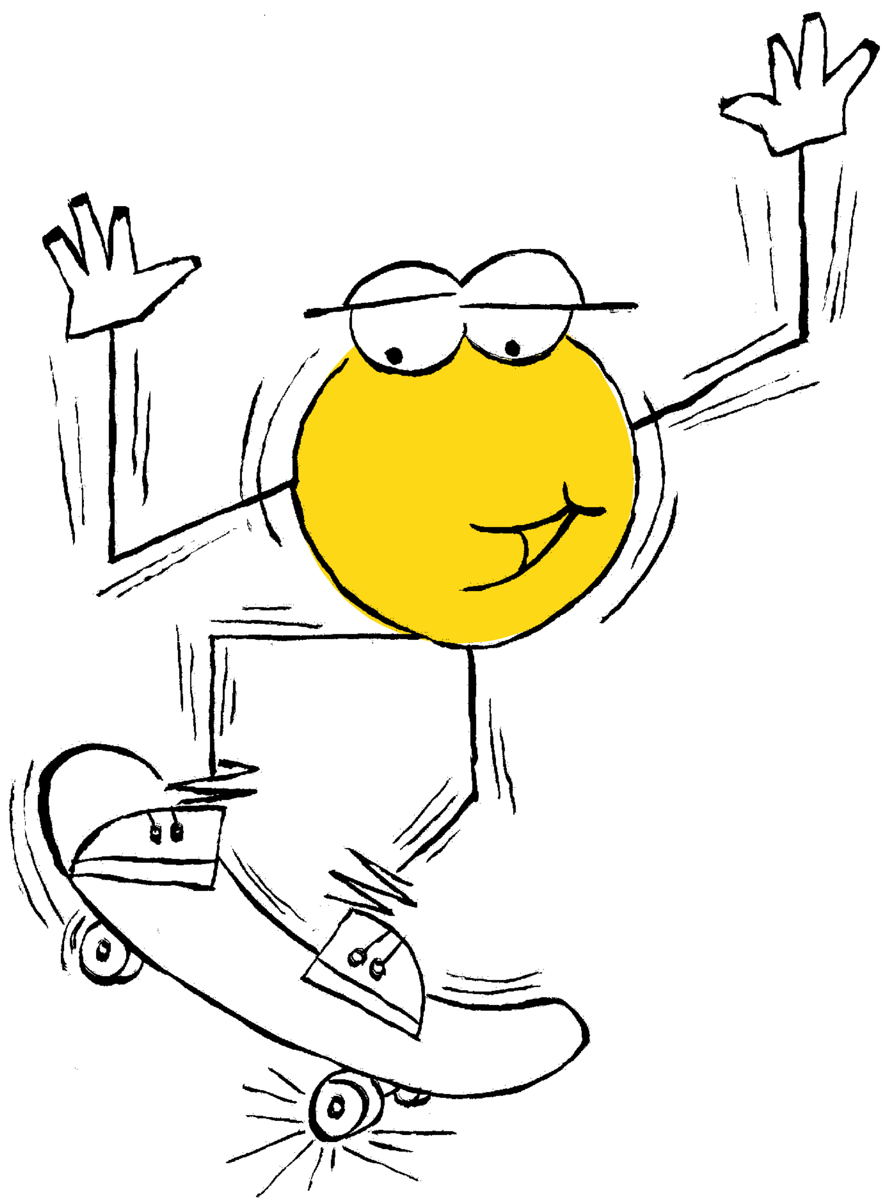
Funcionalidad, analizar si el software ofrece todo lo que un programa de esas características debería ofrecer (en algunos casos en comparación con el programa privativo usado con la misma función).

Tendencia, el desarrollo de software es algo dinámico, es importante tener en cuenta si el software en cuestión mejora continuamente, si la comunidad de desarrolladores es activa y aporta mejoras en forma permanente que hace que dicho software esté al día con la tendencia para esa área de desarrollo.

En el Anexo I se puede encontrar una lista de Software Libre seleccionado usando estos criterios.

COMO MIGRAR A SOFTWARE LIBRE





COMO MIGRAR A SOFTWARE LIBRE

En el contexto de esta guía, estamos encarando el análisis de la migración en el ámbito de pequeñas organizaciones. Si bien muchos consejos son aplicables a cualquier ámbito, no vamos a profundizar en los aspectos técnicos, los cuales varían muy rápidamente y dependen en muchos casos de la infraestructura existente, sino que vamos a hacer énfasis en lo metodológico y las opciones disponibles para encarar la migración. En todo caso, siempre será conveniente contar con ayuda de personal técnico, que en buena medida puede surgir de los integrantes de la misma comunidad de Software Libre.

Aspectos importantes relacionados con la migración:

1) Lo ideal sería partir de cero utilizando Software Libre, pero si estamos hablando de migración ese no es el caso. O sea, estamos partiendo de una situación en la cual utilizamos Software Privativo, y deseamos pasar a Software Libre. Esto conlleva una serie de problemas inherentes a cualquier migración (inclusive de una versión de Software Privativo a otra) y otros inherentes a la migración de Software Privativo a Software Libre, que presenta algunas características especiales.
2) Debemos pensar que una parte del diseño de los sistemas privativos apunta justamente a evitar la migración, lograr la “fidelización forzosa” del cliente, por lo cual siempre tendremos que sortear una serie de inconvenientes para migrar con éxito.
3) Para una migración exitosa, es importante contar con

defensores del cambio dentro del equipo de trabajo y el apoyo de los usuarios del sistema. Para lograr esto, siempre es conveniente, antes de migrar, realizar charlas explicativas informando sobre el por qué del cambio y de los beneficios esperados. Involucrar positivamente a los usuarios del sistema en el cambio siempre es un diferencial importante a la hora de realizar la migración.

4) Realizar un claro diagnóstico de la situación de partida o escenario, el cual incluye la arquitectura del sistema, configuración de la red, hardware y software utilizado. Además de las características de los usuarios del sistema, conocimientos de informática, necesidad de capacitación, etc.

5) Definir con la misma precisión la situación que esperamos encontrar al terminar la migración.

6) Justificar el por qué de la migración, muchos de los argumentos se pueden encontrar en otros capítulos de esta misma guía, también es conveniente realizar un análisis de costos de la migración y proyectar los ahorros factibles, con el correr del tiempo, en pago de licencias.

7) Describir el plan de migración, teniendo en cuenta la forma de migración por la que vayamos a optar (más adelante veremos algunas propuestas).

8) Uno de los elementos más delicados es el procesamiento de los datos existentes, los cuales en muchos casos se encuentran guardados en un formato cerrado privativo, y deben ser convertidos a un formato abierto, lo que en algunos casos es muy sencillo (luego veremos ejemplos). Otra opción,

en caso de que dichos archivos no se puedan convertir, es mantener un equipo dentro de la red funcionando con el software específico para manejar esos archivos. En todo caso deberemos, de ahí en adelante, usar siempre formatos abiertos y estándar en el manejo de la información, ya que ésta es la única forma de asegurar un acceso permanente a dicha información, además de su conservación en mediano y largo plazo.

9) Es fundamental, de acuerdo al tamaño de la organización, contar con un equipo técnico que pueda brindar asesoramiento a los usuarios luego de la migración, evacuar consultas y solucionar problemas que puedan surgir. También es importante contar con cursillos de capacitación previos a la migración y posteriormente a la misma.



FORMAS DE MIGRAR:

1)Paso a paso:

A. La forma más simple de iniciar una migración es comenzar a usar Software Libre en el sistema operativo privativo que estemos utilizando. Dado que el Software Libre es básicamente una forma de licenciamiento, existe Software Libre para todos los sistemas operativos existentes.Por lo tanto, podemos comenzar sustituyendo aplicaciones de software privativo que usamos por sus contrapartidas en el ámbito del Software Libre. El Anexo 2 contiene una tabla con las opciones de Software Libre que sustituyen a las aplicaciones privativas más comunes. Un buen ejemplo de lo anterior está en el área de ofimática, en la que el paquete de Software Libre Open Office ofrece las mismas funciones que sus contrapartes privativas, al tiempo que sustituye los formatos cerrados (.doc, .xls, .ppt) por el formato abierto y estándar OpenDocument.

B. Una de las principales ventajas de esta forma de migración es que se pueden ir pasando de los formatos cerrados (por ejemplo los archivos .doc) a formatos abiertos (como por ejemplo .odt) en forma progresiva, y sin cambiar el entorno operativo. Por otra parte, posibilita el intercambio de archivos con usuarios de Software Libre bajo cualquier sistema operativo.

C. Una vez que los usuarios se han familiarizado con las aplicaciones libres de uso cotidiano, se puede

comenzar la migración del sistema operativo. Eso se puede hacer en forma gradual máquina a máquina y/o por secciones. Una de las ventajas de utilizar archivos con formato abierto y estándar es, justamente, liberarnos de la dependencia a un determinado software sobre un determinado sistema operativo. El utilizar dentro de la red equipos con sistemas operativos diversos tiene algunas implicaciones desde el punto de vista técnico, pero es totalmente factible. Inclusive una de las migraciones más fáciles, desde el punto de vista de los usuarios, es la del servidor, dado que su funcionamiento es transparente para los usuarios.

D. La meta de este cambio gradual es terminar en un sistema basado en un cien por ciento en Software Libre.

2) De una sola vez, sustituyendo de entrada el sistema operativo y todas las aplicaciones por Software Libre. Esta es la opción más rápida, pero la más riesgosa, necesita de personal capacitado muy convencido del cambio y una mayor planificación.

CON RELACIÓN A LOS RECURSOS HUMANOS:

1) Es importante que todo el personal de la institución este informado del tema y sienta que la migración va a generar beneficios en un cierto plazo.

2) Es importante brindar opciones de capacitación para el personal que lo considere necesario, es una forma de vencer el “miedo a lo desconocido” uno de los escollos más frecuentes en el momento de encarar una migración.

3) Es importante que el personal sienta que la migración le abre nuevas puertas en el ámbito de su capacitación personal, dado que el Software Libre es una opción de futuro y una forma de apropiarse realmente de las nuevas tecnologías.

4) Es importante lograr que el personal actual del área de informática esté involucrado y comprometido en el proceso ya que son las personas clave en la migración, ellos deben ser especialmente considerados y los primeros en ser capacitados en el ámbito del Software Libre, en caso de ser necesario dada su formación en el área.

¿QUÉ SE PUEDE HACER DESDE AHORA PARA FACILITAR UNA MIGRACIÓN FUTURA?

Desde este momento, aunque usted aún no esté todavía decidido a migrar a Software Libre, puede tomar algunas medidas para facilitar una migración futura y disminuir su nivel de dependencia con relación al software privativo:

1) Insistir en el uso de Formatos Abiertos y Estándar es la única forma de garantizar que la organización pueda acceder hoy y en el futuro a los datos y la información que maneja. Además, es una base muy importante para facilitar una futura migración y el intercambio de información con los que ya están utilizando Software Libre.

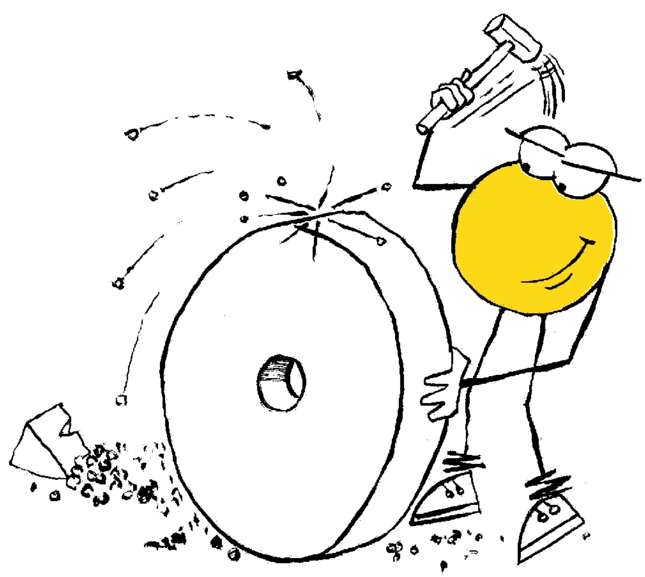
2) Desarrollar sistemas basados en tres niveles, donde el código sea independiente de la interfaz y de los métodos de acceso a los datos.

3) Insistir en que las nuevas aplicaciones generadas sean portables, o sea, usar lenguajes portables como ANSI C, JAVA, Perl, Python, etc. Evitar lenguajes de arquitecturas específicas.

4) Evitar la construcción de aplicaciones que requieran la presencia de otras aplicaciones privativas.

5) Insistir en que cualquier desarrollo web de la organización pueda ser visualizado en los navegadores más comúnmente usados en la web, principalmente los licenciados como libres, (por ejemplo Firefox).

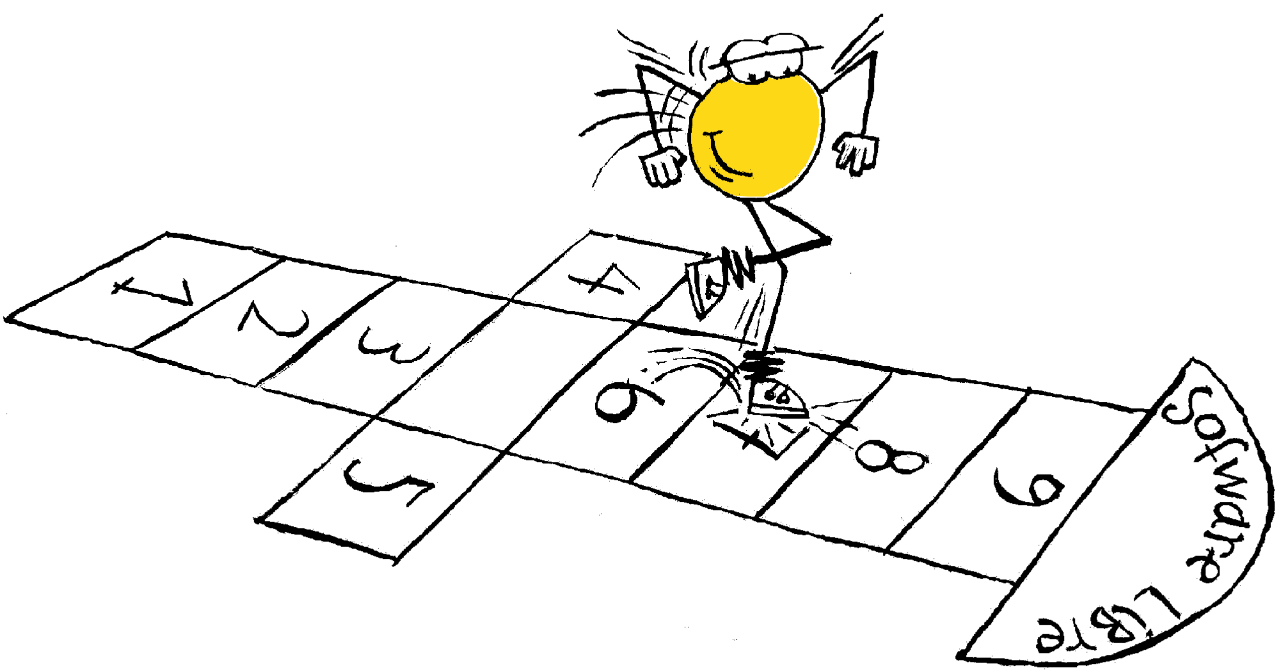
6) Insistir en que los desarrollos web cumplan con los estándares de la W3C. Desarrollar los sitios web de forma tal que sean fácilmente portables a un servidor que corra sobre Software Libre. Un ejemplo: Un sitio web en un servidor que corra sobre Windows no distingue la diferencia entre mayúsculas y minúsculas en el nombre de los archivos, pero en un servidor Apache bajo GNU/Linux esa diferencia se manifiesta, por lo tanto es conveniente organizar el sitio web de tal manera que se tenga en cuenta que un archivo llamado autores.html no es igual a otro llamado Autores.html: los dos serán vistos como archivos diferentes, por lo que al realizar un link a dicho archivo se debe mantener el criterio de llamarlo con mayúscula o minúscula.



Para los interesados en profundizar en los aspectos técnicos de la migración, recomendamos la publicación Materiales de Directrices IDA (Intercambio de Datos entre Administradores) de migración a software de fuentes abiertas, http://www.netproject.com

SOFTWARE LIBRE EN EDUCACIÓN




1
2
3
4
5
6
7
8
9
Software Libre

¿POR QUÉ USAR SOFTWARE LIBRE EN LA EDUCACIÓN?

Existen muchos motivos por los cuales es importante usar Software Libre en la Educación. Tal vez lo fundamental, de acuerdo con la visión de Richard Stallman, sea tener en cuenta al estudiante como un ser integral y enseñarle la importancia de la libertad, guiarlo en el sentido de saber hacer uso de esa libertad. Hacerle entender al estudiante que debe tener la posibilidad de estudiar hasta donde su curiosidad lo lleve, que debe poder profundizar en el conocimiento, y que además existe una responsabilidad inherente a ello, un Software Libre es tan bueno como la responsabilidad con que encararan sus realizadores su trabajo en torno a él.

Pero además existen otras razones, los costos para las instituciones educativas son menores, no solo por el ahorro en el pago de licencias, sino también porque, como ya explicamos anteriormente, el Software Libre es más confiable y seguro que su homólogo privativo.

El poder acceder al código fuente es un estímulo permanente para la apropiación de las nuevas tecnologías y la innovación. Lo cual pasa también por entender que el software se puede adaptar a sus necesidades locales y que cualquier persona, con los conocimientos necesarios, puede participar de la construcción, adaptación y uso de programas de computadora. Esta es una forma real y tangible de acercar las nuevas tecnologías a nuestra gente, acercando las posibilidades, en lugar de mostrar al software como la obra de algún iluminado que seguramente vive en un país desarrollado y necesita una enorme infraestructura para su trabajo. Esto no es así, y el software libre es una muestra tangible de eso.

La posibilidad de copiar y difundir el software, sin incurrir en copias

ilegales, es una forma de evitar que la propia institución educativa lleve a sus alumnos y docentes a violar la Ley, cosa que pasa en muchos ámbitos cuando por razones diversas se estimula la realización de copias ilegales de software privativo. Suele suceder, además, que esto ocurra sin que los docentes, alumnos o familiares vinculados a la escuela tengan siquiera conciencia de que están cometiendo un acto penado por la ley.

Pero además le estamos enseñando al estudiante que hay muchas cosas por hacer, que la obra nunca está acabada y que puede colaborar en su construcción, que su aporte es bienvenido, que puede cooperar con otros, que puede difundir el conocimiento, todo ello participando en la construcción, modificación y difusión del software.

En cuanto a la investigación, tarea fundamental de toda institución educativa, el uso de Software Libre es básico dado que el software privativo nos impide conocer como funciona un programa. Por ejemplo, si en un trabajo de investigación utilizamos software privativo en el procesamiento de los datos, ¿de qué manera podemos hacer la refutación de los resultados obtenidos?, no podemos revisar todos los pasos que llevaron a la obtención de un determinado resultado.Es imposible, dado que el software privativo actúa como una caja negra y no podemos acceder a los algoritmos utilizados para el procesamiento de los datos (no hay acceso al código fuente).

SOFTWARE LIBRE EN LA ADMINISTRACIÓN PÚBLICA





La utilización de Software Libre y Formatos Abiertos y Estándar en la Administración Pública tiene una serie de implicancias que hacen que muchos consideren la necesidad de legislar en la materia, exigiendo que el Estado utilice exclusivamente Software Libre y Formatos Abiertos y Estándar.

Veamos cuales son los puntos en consideración que llevan a plantear esto:

1) **La confiabilidad y seguridad del sistema.**
Cuando el código fuente del software está disponible para que cualquier programador lo pueda estudiar y revisar, es mucho más probable obtener un producto estable y confiable. "Dados muchos ojos, todos los errores serán obvios". A esto se lo bautizó como Ley de Linus – en alusión a Linus Torvalds, el programador finlandés que inició el proyecto Linux-..Cuando se cuenta con el código fuente, el software puede ser auditado para encontrar fallas de seguridad. Esto implica procedimientos de seguridad que se basan en la claridad y transparencia (el conocimiento en profundidad de la fortaleza del algoritmo utilizado y no de la oscuridad, algo que de ser conocido por terceros tira por tierra la seguridad del sistema).

2) **El acceso, control de la información y efecto de red.**
El uso de formatos abiertos y estándar evita que el Estado quede preso de un software específico controlado exclusivamente por una empresa. Por otra parte, asegura el acceso a la información durante toda la vida útil de la misma, cosa imposible de asegurar con formatos cerrados. Tengamos en cuenta que cuando hablamos de la vida útil de los datos, en



Linus Torvalds
Es reconocido mundialmente por haber iniciado el desarrollo del núcleo (Kernel) de Linux (que le debe su nombre). Torvalds nació en Finlandia en el año 1969. Siendo un estudiante de Ingeniería Informática en la Universidad de Helsinki y con apenas un año de experiencia en programación en C, Linus inició un proyecto personal para desarrollar un núcleo similar a Minix. En 1991 publicó la primera versión de Linux. El punto de inflexión de la historia es 1992, cuando Linus Torvalds decide que Linux se distribuya bajo la Licencia GNU/GPL, diseñada por la Fundación del Software Libre. Desde entonces, cientos de programadores de todo el planeta han contribuido para que Linux sea un núcleo sólido y sirva de base para un gran número de distribuciones de Software Libre.
Más información en http://es.wikipedia.org/wiki/Linus_Torvalds

el caso de la Administración Pública, esto no es un tema menor, ya que buena parte de los datos recabados allí deben estar disponibles para nuestra generación tanto como para las generaciones futuras. El uso de formatos cerrados es utilizado por las empresas de software privativo para evitar que los usuarios puedan migrar hacia software de una empresa competidora o hacia Software Libre. Para entender como ocurre esto, tenemos que considerar el Efecto de Red, el cual se basa en la denominada Ley de Metcalfe (Formulada por Robert Metcalfe, uno de los desarrolladores del estándar Ethernet), la que sostiene que “el valor de un sistema de comunicaciones aumenta proporcionalmente al cuadrado del número de usuarios del sistema”. Esto quiere decir que cuanto más usuarios usen un formato de archivos, más útil será este para ellos, dado que será usado para intercambiar información dentro de una gran base de usuarios, y más difícil será cambiar a un formato diferente. Si dicho formato es, además, secreto y propiedad de una empresa en particular, lo anterior le da mucho poder. Cuando el Estado es quien utiliza ese tipo de formato, el problema es aún mayor, dado que es el Estado quien empieza a volverse dependiente de una única empresa.Cuando se trabaja con formatos abiertos y estándar, la dependencia generada por el efecto de red se atenúa. Cualquier empresa puede generar software para trabajar con un formato de archivos abiertos y estándar, dado que sus especificaciones son accesibles libremente. Por otra parte, se estimula la libre competencia entre proveedores. Un buen ejemplo de formato abierto y estándar es el formato OpenDocument.

3) **La transparencia con que se instrumenta todo el sistema de manejo de la información.**

Cuando firmamos un documento en papel, sabemos lo que estamos haciendo: estamos apoyando la pluma sobre el papel respecto del que expresamos conformidad. Se puede adulterar el papel, y la conformidad, pero es difícil hacerlo. Pensemos ahora en un sistema de firma electrónica, la situación es distinta: la máquina nos muestra el documento en pantalla, y nos pide que ingresemos la contraseña de nuestra firma. ¿Cómo podemos saber que lo que la máquina firma es lo que nos mostraba, y no otra cosa? ¿Cómo podemos saber que no firmó “además” otras cosas, sin decírnoslo? La única manera que tenemos de confiar en el proceso es si podemos auditar completamente todo el software, y corregirlo para asegurarnos de que haga lo que dice hacer, y sólo lo que dice hacer.

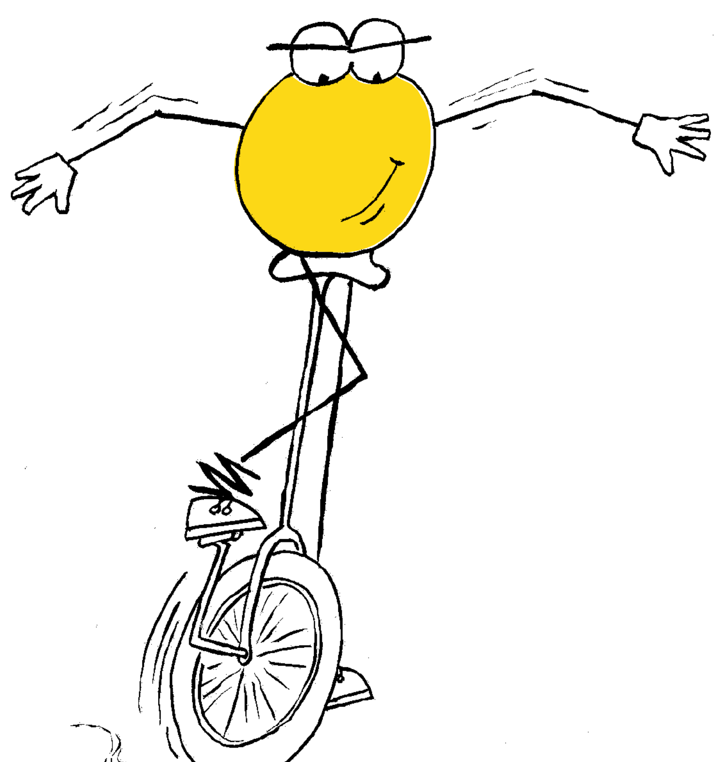
4) **El costo de su implementación.**

El software cuesta dinero. Inclusive el Software Libre no es sinónimo de software gratuito, puede serlo o no. Lo que sí es seguro, es que con el Software Libre el Estado se ahorra el pago de licencias y que, si algo debe pagar, lo hace en desarrollo, mantenimiento y formación. Esto implica que con una perspectiva de futuro, la opción por el Software Libre es la más económica. Cuando el Estado utiliza Software Libre y Formatos Abiertos, puede elegir el proveedor, dado que todos los desarrolladores se encuentran en igualdad de condiciones. Por otra parte, lo que el Estado gasta en desarrollo, mantenimiento, formación, etc, queda en el país, se

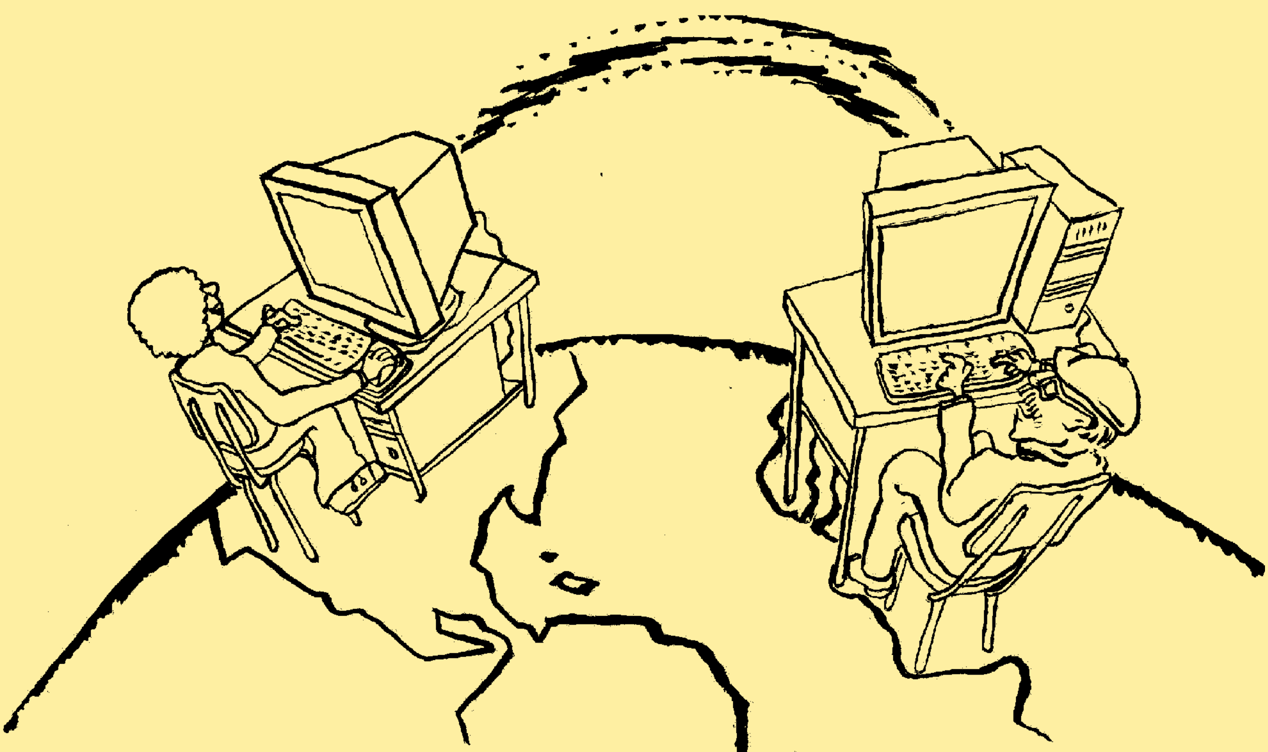
redistribuye, se genera empleo y una parte retorna al propio Estado en el pago de impuestos.

5) **La independencia tecnológica que pueda tener el Estado de los proveedores.**

La apropiación de las nuevas tecnologías implica mucho más que saber utilizar el ratón y el teclado, implica profundizar en la forma en que ha sido desarrollado el software, esto solo es posible con el Software Libre. Cuando se utiliza Software Libre se produce una difusión horizontal del conocimiento, los programadores aprenden y mejoran su trabajo, el software se comparte, se combate la brecha digital. Lo anterior asegura, con el tiempo, una mayor independencia, una apropiación de la tecnología y a la vez un ahorro en costos.



DESARROLLO DE SOFTWARE LIBRE



DESARROLLO DE SOFTWARE LIBRE

Las condiciones de licenciamiento de los programas libres permiten la construcción comunitaria de software. No es necesario inventar la rueda cada vez: los desarrolladores de software pueden acudir a inmensas colecciones de programas y bibliotecas altamente funcionales e intensamente probadas. Esto reduce el esfuerzo y el riesgo de desarrollo, comparado con la alternativa de empezar de cero.

Usando el modo cooperativo de construcción, usamos un mecanismo similar al del peer review (revisión por pares o arbitraje) tan esencial al método científico, y no limitamos las posibilidades del programa a lo que pueda ocurrírsele a un grupo pequeño de usuarios.

Un buen ejemplo de lo anterior es la evolución de la World Wide Web desde la primera versión de Tim Berners Lee (desarrollador del HTML) hasta el presente.

A diferencia de los bienes materiales y escasos, como un campo o una máquina, cuyo valor para cada usuario disminuye proporcionalmente a la cantidad de personas con las que debe compartirlo, el software aumenta en valor mientras más lo compartimos. El efecto de red hace que un programa sea más útil, y por lo tanto más valioso, mientras más gente lo usa, porque es más fácil intercambiar información, experiencias y resultados con usuarios del mismo programa.

Los defensores del software privativo saben esto, y así toleran una gran cantidad de copias ilegales de sus programas, porque saben que esa difusión ilegal es el cimiento del valor de sus productos.

Así, el valor potencial de los programas libres es mayor que el de los no libres, tanto desde el punto de vista social como individual: no hay

restricciones a la difusión del programa, y tampoco a su utilización.

Ha habido excelentes programas que no pudieron imponerse en el mercado aún pese a sus virtudes técnicas hasta que fueron publicados como Software Libre, lo que los hizo enormemente valiosos.

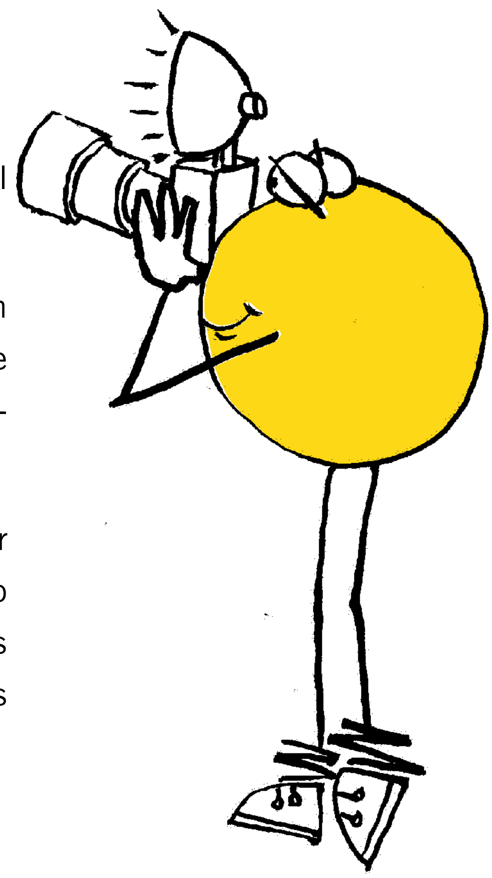
El modelo de negocios del Software Libre no parte de la producción pseudo-industrial de programas para vender como producto terminado, sino en el agregado de valor. Esto posibilita muchos negocios en las áreas de capacitación, asesoramiento, adaptación, documentación, publicación de libros, etc. Existen muchas empresas (Google, ISPs) cuyos servicios no podrían ser ofrecidos con la calidad necesaria y a un precio aceptable si no contaran con Software Libre.

Para desarrolladores de software, el Software Libre ofrece una oportunidad poderosísima: la de agregar valor mediante la ampliación incremental de la funcionalidad de los programas. Los usuarios de software (independientemente de que éste sea libre o no), a menudo se encuentran con que los programas no hacen exactamente lo que ellos necesitan, que hacen cosas de más, o de menos. Si el software no es libre, un desarrollador que quiera satisfacer esta necesidad deberá primero duplicar la funcionalidad del programa que el usuario tiene, y recién entonces agregarle la nueva función. Muy frecuentemente esta alternativa es demasiado costosa o incluso ilegal, y tanto el programador como el usuario quedan insatisfechos.

Si el programa es libre, en cambio, el desarrollador puede simplemente agregar la funcionalidad necesaria al programa ya existente, y cobrar al usuario sólo por el agregado. Esto sí es económicamente viable, y contribuye al círculo virtuoso: un programa más funcional es

más tentador para usuarios potenciales, y mientras más usuarios tengan un programa, más chances hay de que haya usuarios dispuestos a mejorarlo.

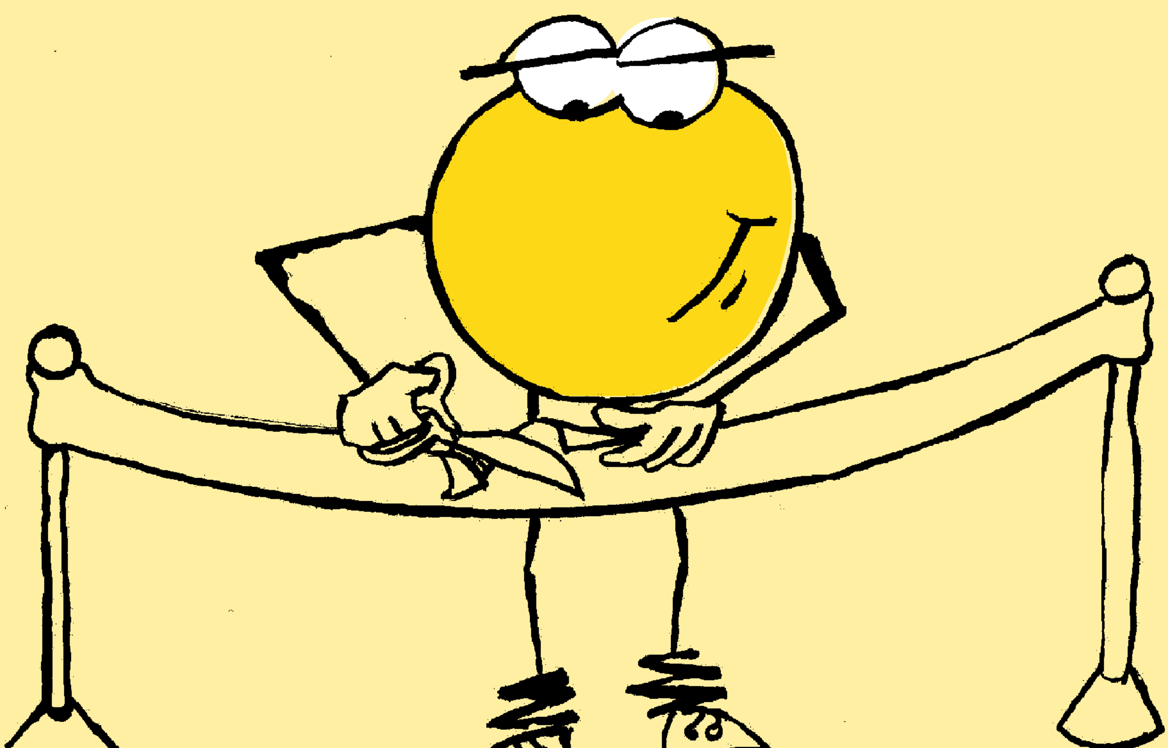
Con todas las virtudes y beneficios del Software Libre, hay iniciativas internacionales que amenazan con restringir su producción, distribución y uso



1. Las patentes de ideas de software pueden hacer ilegal la confección de programas libres para ciertas tareas.
2. La criminalización de la elusión de sistemas de gestión digital de restricciones hace ilegal la confección de programas libres para acceder a información codificada en ciertos formatos.
3. La proliferación de dispositivos que no permiten usar programas modificados, así como de servicios que no son accesibles salvo desde dispositivos con estas características hacen imposible escribir programas libres para acceder a ellos.

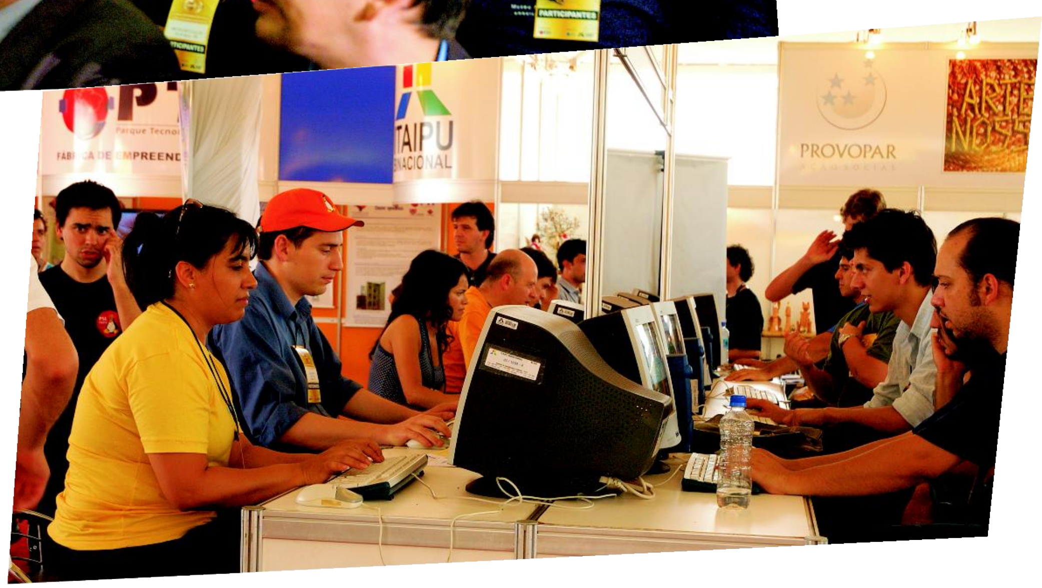
Es imprescindible que los gobiernos de la región y los organismos internacionales tomen conciencia de estos riesgos, para poder seguir usando la informática sin necesidad de entregar a cambio libertad, independencia y soberanía.

EVENTOS DE SOFTWARE LIBRE




LACFree 2005
Software Livre


PARTICIPANTES


PROVOPAR

EVENTOS DE SOFTWARE LIBRE DE AMÉRICA LATINA CON PROYECCIÓN INTERNACIONAL

A lo largo y ancho de América Latina, las diferentes agrupaciones de usuarios, fundaciones, incubadoras de empresas relacionadas al Software Libre y en algunos casos los propios Estados, organizan eventos con el propósito de acercar a las empresas, usuarios y desarrolladores los principales valores de éste movimiento. De estos eventos participan disertantes locales, invitados de la región y reconocidas figuras del Software Libre a nivel mundial.

De los muchos eventos de Software Libre realizados anualmente en la región, algunos de los más importantes son:

- CaFeConf. El evento anual organizado por el Grupo de Usuarios de Software Libre de Capital Federal (CABA-Argentina). Un evento que ha ido creciendo con los años, hasta convertirse en un referente en toda la región. Este año van para la 7ma edición (2007).(http://www.cafeconf.org.ar)

- Congreso Nacional de Software Libre (CONSOL) México, un evento anual que realiza la comunidad de Software Libre de México y que ya va por su sexta edición (Febrero de 2007). Ver www.consol-.org.mx

- FISL (Forum Internacional de Software Libre) Realizado en la ciudad de Porto Alegre(RS-Brasil). Ya está en la 8va edición (año 2007) y se posiciona como uno de los eventos de mayor asistencia en América Latina.(http://www.fisl.org.br)

- FLISoL (Festival Latinoamericano de Instalación de Software Libre) Un evento iniciado en Colombia, y que ha ganado popularidad en todo América Latina. Se trata de una fiesta de instalación de Software Libre coordinada en ciudades de toda América Latina, el mismo día. Tiene como particularidad que además se realiza una-

coordinación conjunta para establecer los pasos a seguir y las pautas generales. (http://www.installfest.info)

- **Foro Mundial de Tecnologías Libres**, un evento que tiene su sede en Venezuela y convoca entusiastas tanto del software libre como de la cultura libre en general.



- **Jornadas Regionales de Software Libre.** Este evento en particular se caracteriza por ser un evento itinerante, nacido en la ciudad de Rosario(Santa Fe-Argentina), pasando por Montevideo en tres ocasiones y luego por Mendoza y Córdoba para su 7ma edición, 2007. (http://jornadas.grulic.org.ar)

- **LACFree (Conferencia Latinoamericana y del Caribe sobre desarrollo y uso de Software Libre)** evento fomentado por UNESCO, la primera reunión fue realizada en Cuzco en 2003, la próxima se prevee para el 2008. El objetivo de la conferencia es promover el intercambio de experiencias en la aplicación del Software Libre en beneficio del desarrollo económico y social. (http://www.unesco.org.uy/)

- **Latinoware,** Conferencia Latinoamericana de Software Libre. Evento organizado con el apoyo del Gobierno del Estado de Paraná Brasil, se realiza anualmente desde el 2003.Se desarrolla generalmente todos los años durante el mes de noviembre en Foz

de Iguazu, Parana Brasil. En el Parque Tecnológico de ITAIPU. http://www.latinoware.org/

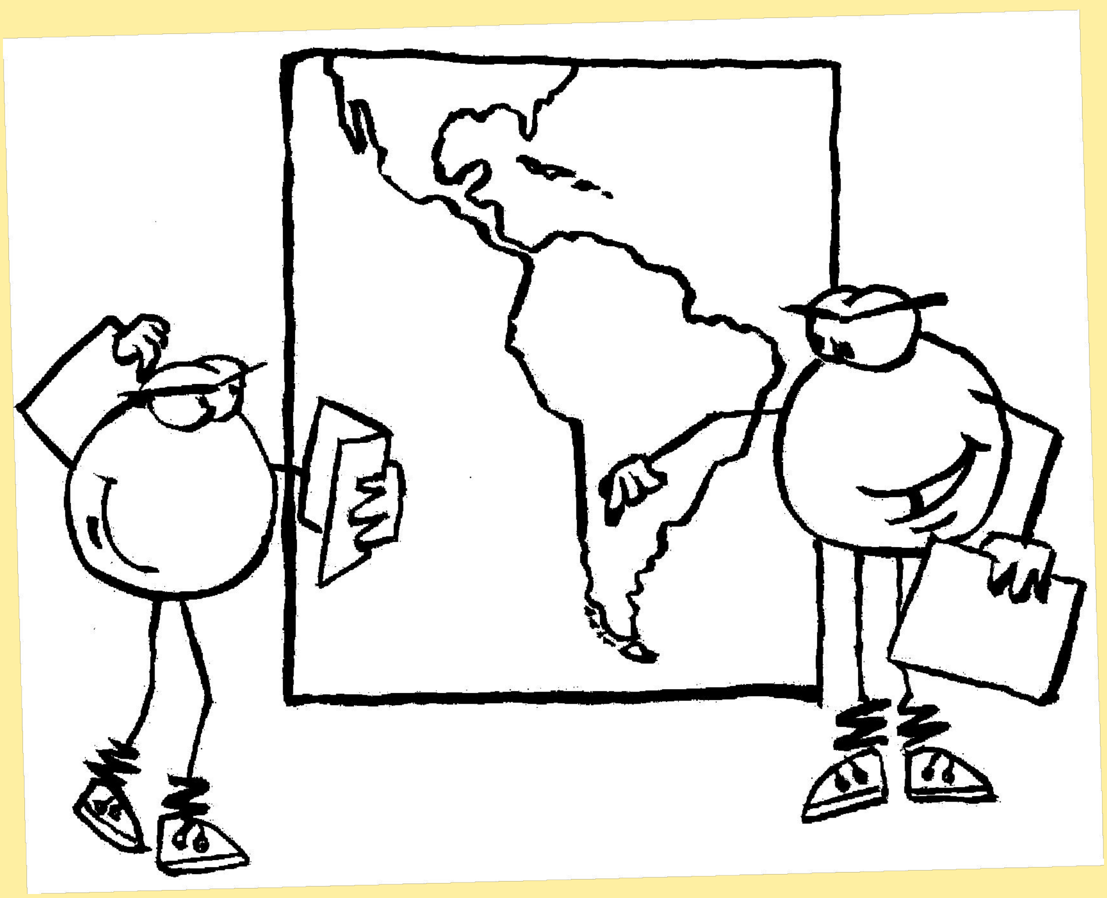


Además de estos eventos, existen eventos más pequeños, pero no por eso menos importantes, como los que organizan algunas universidades, grupos de ciudades más pequeñas, o grupos de interés. El fin de esos eventos es promover el uso de herramientas libres, educar a los usuarios que residen en sus zonas de influencia o comparten intereses comunes.

Este tipo de eventos, como las reuniones periódicas de los grupos, son los pilares de la comunidad y permiten, a la larga, concentrar en determinados puntos de nuestro continente eventos de mayor porte.



ORGANIZACIONES Y PUNTOS DE CONTACTO



ORGANIZACIONES Y PUNTOS DE CONTACTO

Una de las claves para acercarse al Software Libre es tomar contacto con sus comunidades. Esto no sólo sirve para compartir tanto programas como una cultura propia de las comunidades, sino también para saber a dónde recurrir cuando hace falta hacer alguna consulta o resolver algún problema. La sugerencia de esta guía es que usted recurra a una organización, empresa o grupo de usuarios en su región; una de las ventajas del Software Libre es estimular el trabajo local, de la propia comunidad. No hay mejor forma de volcarse al Software Libre que hacerlo a través de sus grupos de usuarios, verdaderos motores de la comunidad.

A continuación una lista, no definitiva, de organizaciones integrantes de la comunidad del Software Libre en los distintos países de América Latina y el Caribe:

ARGENTINA

ANT Rosario- Asociación de Nuevas Tecnologías
http://www.ant.org.ar

Asociación Civil Software Libre Argentina
http://www.solar.org.ar/

Gleducar Comunidad educativa basada en el principio de Construcción Cooperativa de Conocimiento http://www.gleducar.org.ar

Fundación Vía Libre http://www.vialibre.org.ar/
Organización civil sin fines de lucro radicada en Córdoba, Argentina, que se propone contribuir a una adopción de nuevas tecnologías basada en el beneficio para una sociedad libre, democrática y pluralista, en contraste a la tendencia actual de incorporarlas de manera irreflexiva y sin prestar mayor atención a los valores civiles. En esta tarea, el software libre es una herramienta indispensable.

Proyecto Nómade (Comunidad de Artistas y Software Libre)
http://www.nomade.liminar.com.ar

Usuarios de Software Libre de Argentina
www.usla.org.ar
USLA reúne en su sitio web una base de datos completa con la gran mayoría de los grupos de usuarios del país. Es el recurso ideal para encontrar grupos de usuarios de Software Libre en cualquier lugar de Argentina.
Allí está la lista de Grupos de Usuarios más actualizada en
http://www.usla.org.ar/modules/myslugs/

BELIZE

Software Freedom in Belize
http://www.linux.bz/

BOLIVIA

Comunidad de Software Libre Bolivia
http://www.softwarelibre.org.bo/

BRASIL

Abrasol Asociación Brasileña de software Libre
http://www.abrasol.org.br

BrOffice http://www.openoffice.org.br/

Comitê de Incentivo a Produção do Software GNU e Alternativo
http://www.cipsga.org.br/

Incubadora Virtual de Projetos em Software Livre
http://codigolivre.org.br/

Movimento Software Livre do Paraná
http://www.softwarelivreparana.org.br

PHP Brasil http://phpbrasil.com/

Portal do Software Público Brasileiro
http://www.softwarepublico.gov.br/spb/

Projeto Mozilla Brasil http://mozilla.org.br/

Projeto Software Libre http://www.softwarelivre.org/

Proyecto Software Libre Mulheres http://mulheres.softwarelivre.org/

Seminário de Desenvolvimento em Software Livre
http://www.sdsl.org.br/

UnderLinux http://www.underlinux.com.br/

COLOMBIA

Comunidad de Usuarios de Software Libre en Colombia
http://bachue.com/colibri/

COSTA RICA

Grupo de Usuarios de Linux de Costa Rica http://gulcr.org/

CUBA

Portal del Software Libre en Cuba http://www.softwarelibre.cu/

Linux Habana (Grupo de Usuarios de la Universidad de la Habana)
http://www.linuhx.uh.cu/

CHILE

Abrasol (Educación e historia basados en SL) http://www.abrasol.cl/

Centro de Difusión de Software Libre http://www.cdsl.cl

CSoL (Centro de Software Libre)

ChileForge http://chileforge.cl/

Educalibre http://www.educalibre.cl

Software Libre Chile http://www.softwarelibre.cl/

ECUADOR

Asociación Software Libre de Ecuador http://www.asle.ec

Ecuador Linux User Group http://www.ecualug.org/

Open Ecuador http://www.openecuador.org/

EL SALVADOR

Asociación de Usuarios GNU/Linux de El Salvador
http://www.linux.org.sv/

Plaza Linux http://www.plazalinux.com/

GUATEMALA

Free Software Organization Guatemala
http://ong.tupatrocinio.com/free-software-organization-guatemala-ong-400.html

MÉXICO

Comunidad GNU/Linux
http://www.linux.org.mx/

Fundación de Software Libre de México http://www.fsl.org.mx

NICARAGUA

Asociación Social Tecnológica Sé Libre Nicaragua
http://www.selibrenic.org/

PARAGUAY

Comunidad del Software Libre del Paraguay
http://www.linux.org.py/

PERÚ

Asociación Peruana de Software Libre http://www.apesol.org/

Grupo de Usuarios de Software Libre Somos Libres
http://www.somoslibres.org/

PLUG
http://www.linux.org.pe

REPÚBLICA DOMINICANA

Fundación Código Libre Dominicano
http://www.codigolibre.org/

URUGUAY

Debian de Uruguay http://www.debianuruguay.org

LinuxPay - Grupos de Usuarios GNU/Linux de Paysandú
http://www.linuxpay.org

LinuxTeros - Grupo de Desarrollo de Software Libre
http://linuxteros.codigolibre.net

LinuxUruguay - Comunidad de Usuarios Linux del Uruguay
http://www.linuxuruguay.org

Montevideo Libre / Redes Libres
http://www.montevideolibre.org

Portal de la comunidad de Software Libre de Uruguay
http://www.comunidad.org.uy

UyLUG - Grupo de Usuarios Linux de Uruguay
http://www.linux.net.uy

ORGANIZACIONES INTERNACIONALES

Free Software Foundation
http://www.fsf.org/

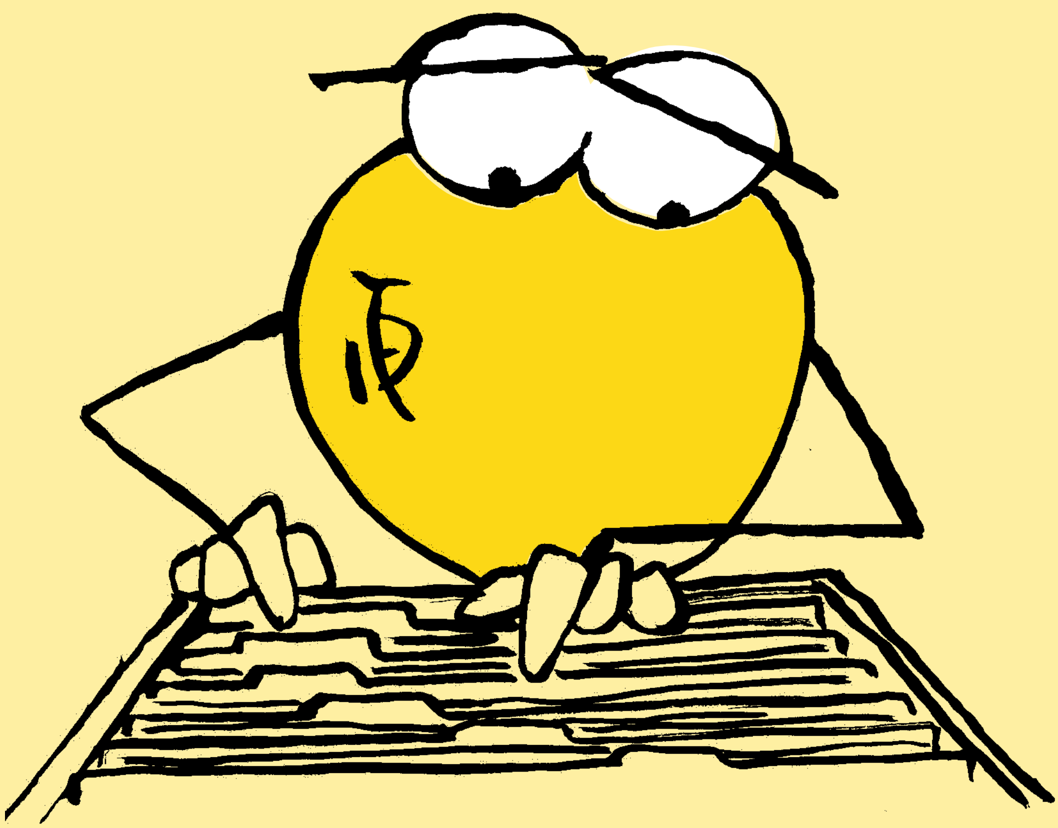
Free Software Foundation Europa
http://www.fsfeurope.org/

FSFLA - Fundación Software Libre América Latina
http://www.fsfla.org/

UNESCO Free & Open Source Software Portal
http://www.unesco.org/webworld/portal_freesoft

UNESCO Free Software Portal
http://portal.unesco.org/ci/ev.php?URL_ID=12034&URL_DO=DO_TOPIC
&URL_SECTION=201

ANEXOS



ANEXO I

La siguiente lista de programas es simplemente una muestra de Software Libre basada en los parámetros definidos en la sección¿Cómo seleccionar Software Libre?.

No es una lista exhaustiva, ni final, busca ser simplemente una introducción a la amplia gama de opciones que ofrece el software libre. Si bien todos los programas abajo mencionados no han sido licenciados bajo la licencia GPL, todos respetan las cuatro libertades que definen al software libre.

En los casos en que fue posible hacerlo se incluyen, además de los links al sitio oficial, links a sitios web en español.

Aplicaciones destinadas a la empresa			
Producto	**Descripción**	**URL**	**Licencia**
OTRS (Open-source Ticket Request System)	Sistema de mensajes (tickets) para soporte	http://otrs.org/	GPL
SugarCRM	Manejo de marketing, ventas, soporte de clientes, etc.	http://www.sugarcrm.com	

Bases de Datos			
Producto	**Descripción**	**URL**	**Licencia**
Firebird	Base de datos relacional	http://www.firebirdsql.org/	MPL 1.0
Mysql	Base de datos relacional	http://www.mysql.com/ http://dev.mysql.com/doc/refman/5.0/es/	GPL
PostgreSQL	Base de datos relacional	http://www.postgresql.org/ http://www.postgresql.cl/	BSD

Comunicación			
Producto	**Descripción**	**URL**	**Licencia**
Evolution	Manejador de correo, calendario y manejador de direcciones para GNOME	http://www.gnome.org/projects/evolution/	GPL
Firefox	Navegador Web	http://www.mozilla-europe.org/es/	MPL
Gaim	Cliente de Mensajería Instantánea	http://gaim.sourceforge.net/ http://www.jabberes.org/clientes/gaim/	GPL
Kontact	Cliente de PIM y correo para KDE	http://kontact.kde.org/kmail/	MPL
Konqueror	Navegador web y manejador de archivos para KDE	http://www.konqueror.org/	GPL
SpamAssassin	Filtro antispam	http://spamassassin.apache.org/	GPL
Thunderbird	Manejador de correo	http://www.mozilla-europe.org/es/products/thunderbird/	Apache

Gestión de Cursos

Producto	Descripción	URL	Licencia
Moodle	Permite el manejo de cursos, participantes, examenes, autoevaluación, chats, foros, etc.	http://www.moodle.org http://docs.moodle.org/es/Portada	GPL

Interfaces gráficas

Producto	Descripción	URL	Licencia
GNOME	Entorno gráfico de escritorio para LINUX	http://www.gnome.org/ http://www.es.gnome.org/	GPL
KDE	Entorno gráfico de escritorio para LINUX	http://www.kde.org/ http://www.kdehispano.org/	GPL

Ofimática

Producto	Descripción	URL	Licencia
GIMP	Programa de edición de imagen	http://www.gimp.org/ http://gimp.org.es/	GPL
Inkscape	Dibujo vectorial	http://www.inkscape.org/ http://www.inkscape.org/doc/basic/tutorial-basic.es.html	GPL
Koffice	Suite de oficina, integrada, para KDE	http://www.koffice.org/	LGPL
OpenOffice	Suite de oficina, completa. Usa el formato estándar OpenDocument.	http://es.openoffice.org/	GPL
Scribus	Programa de autoedición	http://www.scribus.net/	GPL

Para trabajo colaborativo

Producto	Descripción	URL	Licencia
eGroupware	Manejador de correo, calendario, manejador de contenidos, foros, etc.	http://www.egroupware.org/	GPL
Zimbra	Todo lo necesario para organizar un sistema de trabajo colaborativo. Incluye antispam y antivirus.	http://www.zimbra.com/	MPL

Servidor de Correo y agregados			
Producto	**Descripción**	**URL**	**Licencia**
ClamAV	Antivirus para de correo electrónico para servidores LINUX	http://www.clamav.net/	GPL
Mailman	Administración de listas de correo	http://www.gnu.org/software/mailman	GPL
Postfix	Servidor de correo	http://www.postfix.org/	IBM Public License
Postgrey	Antispam	http://postgrey.schweikert.ch/	GPL
Sendmail	Servidor de correo más utilizado	http://www.sendmail.org/	Sendmail License

Servidor Web			
Producto	**Descripción**	**URL**	**Licencia**
Apache	Servidor web, el más usado a nivel mundial	http://httpd.apache.org/ http://httpd.apache.org/docs/2.0/es/	Apache

Sistemas manejadores de contenido (CMS)			
Producto	**Descripción**	**URL**	**Licencia**
Dokuwiki	Manejador de contenidos basado en el esquema Wiki, no necesita de base de datos.	http://wiki.splitbrain.org/wiki:es:dokuwiki	GPL
Joomla!	Manejador de contenido derivado de Mambo	http://www.joomla.org	GPL
		http://www.joomlahispano.org/	
Mambo	Manejador de contenido	http://www.mamboserver.com	GPL
		http://www.portalmambo.com/	
PostNuke	Manejador de contenido	http://www.postnuke.com	GPL
		http://www.dev-postnuke.com/	
SPIP	Manejador de contenido muy flexible en cuanto a diseño	http://www.spip.net/es	GPL
Twiki	Manejador de contendio basado en el esquema Wiki	http://www.twiki.org/	GPL
WordPress	Manejador de contenidos	http://www.wordpress.org	GPL
		http://code.google.com/p/wordpress-es-xx/	
Xwiki	Manejador de contenidos	http://www.xwiki.org	LGPL

Sistemas Operativos			
Producto	**Descripción**	**URL**	**Licencia**
Debian GNU/Linux	Una de las distribuciones de LINUX más difundida	http://www.debian.org/ http://www.esdebian.org/	GPL
gNewSense	Distribución basada en Ubuntu, con énfasis más que en la funcionalidad en ser 100% libre.	http://www.gnewsense.org/	GPL
Kubuntu	Versión de Ubuntu para KDE	http://www.kubuntu-es.org/ http://www.kubuntu-es.org/	Varias
openSUSE	Distribución no comercial de SUSE, LINUX	http://www.opensuse.org/ http://groups.google.es/group/SuSE-es	Varias
Ubuntu	Distribución con énfasis en la usabilidad y facilidad de uso	http://www.ubuntu.com/ https://wiki.ubuntu.com/LoCoTeamList http://ubuntu.com.es/	Varias

Varios			
Producto	**Descripción**	**URL**	**Licencia**
CUPS	Sistema de impresión común de Unix	http://www.cups.org/	LGPL
Samba	Permite conectividad en redes con Windows	http://www.samba.org http://adminsredes.com/AdminLinux/Samba.php	GPL
Wine	Permite correr sobre LINUX programas desarrollados para Windows	http://www.winehq.org	LGPL

Sitios de referencia para encontrar software libre y/o equivalencias entre Software Libre y Software Privativo:

http://alts.homelinux.net/

http://clic.xtec.net/es/index.htm

http://directory.fsf.org/

http://gnuwin.epfl.ch/apps/es/index.html

http://sourceforge.net/

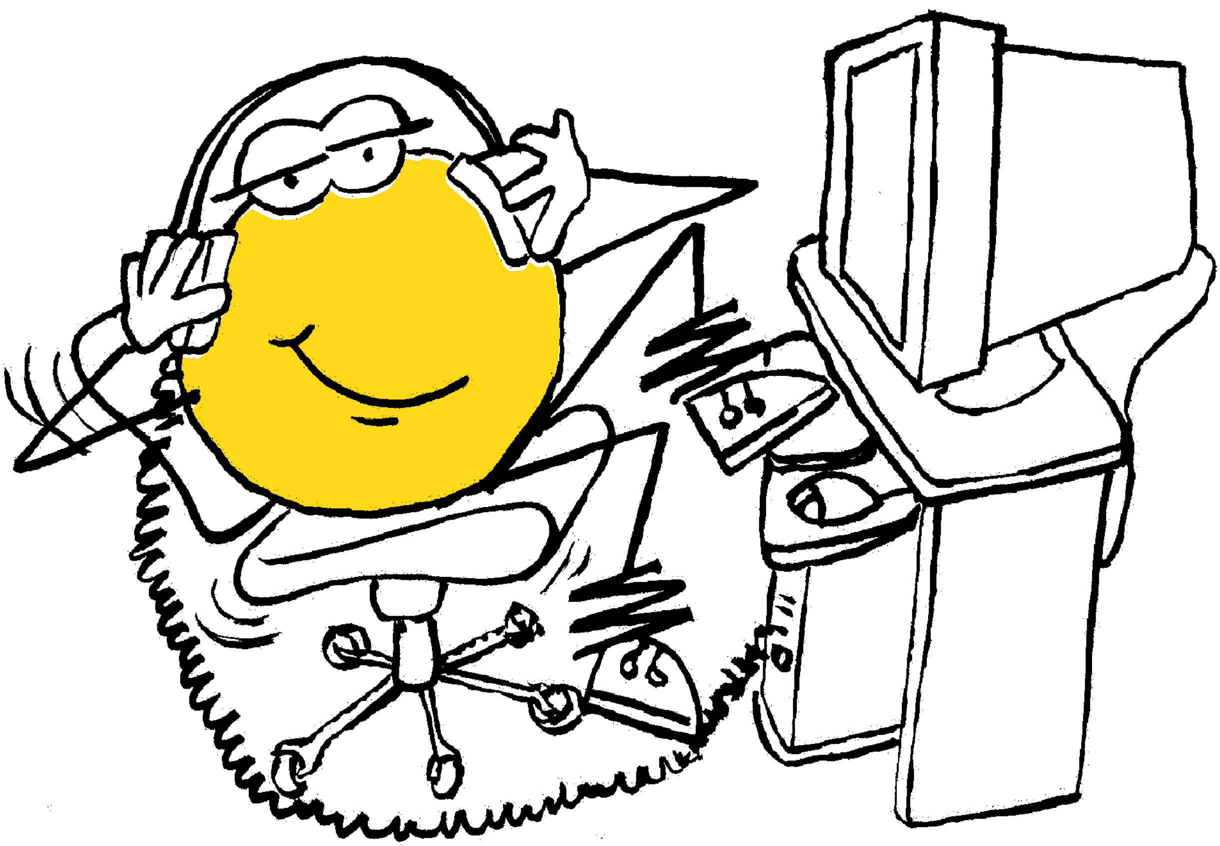
http://winslow.aditel.org/

http://www.cdlibre.org/

http://www.optaros.com/en/publications/white_papers_reports/open_source_catalogue_2007

http://www.osalt.com/

http://www.unesco.org/cgi-bin/webworld/portal_freesoftware/cgi/page.cgi?d=1



ANEXO II

Tabla de programas para Linux, sustitutivos de su versión privativa más difundida.

La mayoría de estos programas, a diferencia, de los presentes en el Anexo I, no han sido analizados, si se ha corroborado la vigencia de los links puestos como referencia.

1) Científicos y Programas Especiales

Descripcion del programa, tareas ejecutadas	Windows	Linux
CAD/CAM/CAE	Autocad, Microstation Mathcad	1) Varkon. (http://www.tech.oru.se/cad/varkon/) 2) Cycas. (http://www.cycas.de/) 3) Tomcad. (http://www.sedoparking.com/tomcad.org) 4) Thancad. (http://thancad.sourceforge.net/) 5) Fandango. (http://www.soffernet.com/jaime/fandango/) 6) Lignumcad. (http://lignumcad.sourceforge.net/doc/en/HTML/index.html) 7) QSCad. (http://qs.co.nz/Tcl/QSCad/) 8) FreeEngineer. (http://freeengineer.org/) 9) Ocadis. (http://www.nocrew.org/software/ocadis/) 10) PythonCAD. (http://www.pythoncad.org/)
CAD/CAM/CAE, simplificado	ArchiCAD	Qcad. (http://www.qcad.org/qcad.html) Es un programa de diseño asistido por ordenador (CAD) de código libre para diseño 2D.
Diseño de esquemas electrónicos	PCAD	1) Eagle. (http://www.cadsoft.de/) Diseña circuitos impresos y realiza esquemas electrónicos. 2) Geda.(http://www.geda.seul.org/)El proyecto gEDA trabaja en la creación de un paquete con licencia GPL que incluye herramientas para diseño de circuitos, captura de esquemáticos (SCH), simulación (SPICE), creación de placas de circuito impreso (PCB), etc Xoscope.(http://linux.maruhn.com/sec/xoscope.html).
Diagramación y Diseño Cartográfico	Microsoft Visio	1) Kivio (Koffice). (http://www.koffice.org/) 2) Dia. (http://www.gnome.org/projects/dia/) 3) Kchart. (http://www.koffice.org/kchart/) 4) xfig. (http://www.xfig.org/) 5) Tgif + dotty. 6) Tulip. (http://www.labri.fr/perso/auber/projects/tulip/)
Editor de Ecuaciones matemáticas	Mathtype, MS Equation Editor	1) OpenOffice Math. (http://www.openoffice.org/) 2) MathMLed. (http://www.newmexico.mackichan.com/mathml/mathmled.htm) 3) Kformula (Koffice). (http://www.koffice.org/) 4) LyX. (http://www.lyx.org/)
Emuladores de circuitos	Electronic Workbench	1) Geda. (http://geda.seul.org/) 2) Xcircuit. (http://bach.ece.jhu.edu/~tim/programs/xcircuit/) 3) Gnome Assisted Electronics. (http://gael.sourceforge.net/) 4) SPICE. (http://embedded.eecs.berkeley.edu/pubs/downloads/spice/index.htm) 5) SPICE OPUS. (http://www.fe.uni-lj.si/~spice/welcome.html) 6) NG-SPICE. (http://ngspice.sourceforge.net/)
Emulador de osciloscopio	Winoscillo	Permite utilizar la tarjeta de sonido como un osciloscopio.
GIS (Sistema de información Geográfica)	ArcView	1) Grass. (http://grass.itc.it/) 2) Quantum GIS. (http://www.qgis.org/) 3) PostGIS. (http://postgis.refractions.net/)
Medida de la temperatura y voltajes sobre placa madre	MBMonitor, PCAlert	KHealthCare (KDE). (http://www.linuxsoft.cz/en/sw_detail.php?id_item=1267)

Descripcion del programa, tareas ejecutadas	Windows	Linux
Para testeo y pruebas HDD	SiSoft SANDRA, SiSoft SAMANTHA	1) hdparm. (http://es.gentoo-wiki.com/HOWTO_Hdparm) Permite visualizar y modificar los parámetros del disco duro para los discos IDE en Linux, de forma que es posible mejorar el rendimiento. 2) Bonnie++. (http://www.coker.com.au/bonnie++/) 3) Iozone. (http://www.iozone.org/) 4) Dbench. (http://samba.org/ftp/tridge/dbench/) 5) Bonnie. (http://www.textuality.com/bonnie/) 6) IO Bench. (no encontrada) 7) Nhfsstone. (http://nfs.sourceforge.net/)
Pequeños Sistemas de Publicación apoyada por el ordenador	MS Publisher	1) Scribus - Desktop Publishing para Linux.(http://www.scribus.net/) Permite diseñar todo tipo de publicaciones, combinado textos, imágenes, formas y colores . 2) Kword. (http://www.koffice.org/kword/) Procesador de texto
Programas de dibujo de estructuras químicas	Chemdraw, Isisdraw	Xdrawchem. (http://sourceforge.net/projects/xdrawchem/)
Programas para observar temperaturas, fanspeeds, etc	SiSoft SANDRA, SiSoft SAMANTHA	1) Ksensors. (http://ksensors.sourceforge.net/ 2) Lm_sensors. (requiere autorización)
Sistemas Matemáticos al estilo MathCad	MathCad	Gap. (http://www.gap-system.org/) Es un sistema de álgebra computacional discreta
Sistemas Matemáticos al estilo Matlab	Matlab	1) Matlab para Linux. [FTP] (http://www.mathworks.com/products/matlab/requirements.shtml) 2) Octave. (+ Gnuplot) (http://www.gnu.org/software/octave/) 3) Scilab. (http://www.scilab.org/) 4) Yorick.(http://web.mit.edu/afs/athena/software/yorick_v1.5.12/yorick/1.5/doc/) Es un lenguaje de programación interpretado, para cálculos y simulaciones científicas. 5) Yacas. (http://www.xs4all.nl/~apinkus/) 6) Euler. (http://euler.sourceforge.net/)
Sistemas Matemáticos al estilo Mathematica	Mathematica	1) Mathematica for Linux. (http://www.wolfram.com/products/mathematica/index.html) 2) Maxima. (http://maxima.sourceforge.net/es/) Es un sistema para la manipulación de expresiones simbólicas y numéricas, incluyendo diferenciación, integración, expansión en series de Taylor, transformadas de Laplace, ecuaciones diferenciales ordinarias, sistemas de ecuaciones lineales, y vectores, matrices y tensores. 3) MuPad. (http://www.mupad.de/) Es un programa comercial de álgebra computacional
Sistemas Matemáticos al estilo Maple	Maple	1) Maxima. (http://maxima.sourceforge.net/) 2) MuPad. (http://www.mupad.de/)
Sistema de tratamiento de texto Avanzado en estilo TeX	MikTex, emTeX (DOS)	1) TeTeX / LaTeX (http://www.latex-project.org/) 2) LyX (WYSIWYM).(http://www.lyx.org/) 3) Kile. (http://kile.sourceforge.net/)

1) Científicos y Programas Especiales

Descripcion del programa, tareas ejecutadas	Windows	Linux
S.M.A.R.T-atributos y temperatura del disco Duro	ome on CD with mainboard, Active SMART	1) smartctl. (http://smartmontools.sourceforge.net/man/smartctl.8.html) 2) Hddtemp-0.3. (http://www.guzu.net/) 3) IDEload-0.2. (http://www.guzu.net/linux/various.php) 4) Smartsuite-2.1.(http://www.linux-ide.org/smart.html) 5) Smartmontools.(http://smartmontools.sourceforge.net/) 6) Smartsuite. (http://sourceforge.net/projects/smartsuite/)
Software CNC, controlador de herramientas de máquinas	OpenCNC [Prop]	EMC.(http://linuxcnc.org/)
Software Econométrico	Eviews, Gretl	1) Gretl. (http://gretl.sourceforge.net/gretl_espanol.html) Gretl es un paquete de software para realizar análisis econométricos que se utiliza en varios Departamentos de Economía de universidades de todo el mundo.
Testeo de Memoria	SiSoft SANDRA	Memtest86. (http://www.memtest86.com/)

2) Multimedia (audio / CD)

Consola de música / mp3 / ogg players	mpg123 para Windows, dosamp	1) mpg321. (http://mpg321.sourceforge.net/) 2) Orpheus. (http://thekonst.net/en/orpheu
Creadores de Música	Cakewalk, FruityLoops	1) RoseGarden. (http://www.rosegardenmusic.com/) 2) Ardour. (http://ardour.org/)
Editores de Audio	SoundForge, Cooledit, ...	1) Sweep. (http://www.metadecks.org/software/sweep/) 2) WaveForge. (http://www.tfm.ro/waveforge/) 3) Audacity. (http://audacity.sourceforge.net/) 4) Ecasound. (http://ecasound.seul.org/)
Editor de Notas Musicales	Finale, Sibelius, SmartScore	1) LilyPond. (http://lilypond.org/web/) 2) Noteedit. (http://developer.berlios.de/projects/noteedit) 3) MuX2d. (http://mux2d.sourceforge.net/)
Grabación de Cds	Windows Media Player, AudioGrabber, Nero,VirtualDrive, VirtualCD, ...	1) Grip. (http://www.nostatic.org/grip/) 2) Audacity. (http://audacity.sourceforge.net/) 3) tkcOggRipper. (http://www.thekompany.com/projects/tkcoggripper/) 4) Gnome Òoaster. (http://gnometoaster.rulez.org/) 5) Cdparanoia. (http://www.xiph.org/paranoia/) 6) Cd2mp3. (http://www.jamiesdomain.org.uk/) 7) Dagrab. (http://web.tiscalinet.it/marcellou/dagrab.html) 8) SimpleCDR-X. (http://ogre.rocky-road.net/cdr.shtml) 9) RatRip. (http://www.linuxrat.org/projects.html) 10) AutoRip. (http://sourceforge.net/projects/autorip/)
Mezcladores de Sonidos	sndvol32	1) Opmixer. (http://optronic.sourceforge.net/sites/projects.php) 2) Mixer_app (for WindowMaker)
Procesadores de Audio Multitrack	Cubase	1) Ecasound. (http://ecasound.seul.org/)

Descripcion del programa, tareas ejecutadas	Windows	Linux
Programas para quemar CD con GUI	Nero, Roxio Easy CD Creator,...	1) K3b. (KDE). (http://www.k3b.org/) 2) XCDRoast. (http://www.xcdroast.org/) 3) Eclipt Roaster. (http://sourceforge.net/projects/eroaster/) 4) Gnome Toaster. (http://gnometoaster.rulez.org/) 5) CD Bake Oven. (http://cdbakeoven.sourceforge.net/) 6) SimpleCDR-X. (http://ogre.rocky-road.net/cdr.shtml) 7) GCombust. (http://www.abo.fi/~jmunsin/gcombust/)
Programas para recepción de estaciones de radio	VC Radio, FMRadio, Digband Radio	1) cRadio. (http://web.archive.org/web/20060425031955/http://www.leg.uct.ac.za/cRadio/) 2) RadioActive. (http://cactus.rulez.org/projects/radioactive/) 3) XMMS-FMRadio.(http://silicone.free.fr/xmms-FMRadio/) 4) Gqradio. (http://gqmpeg.sourceforge.net/radio.html)
Reproductores Música / mp3 / ogg s	Winamp	1) XMMS (X multimedia system). (http://www.xmms.org/) 2) Noatun. (http://noatun.kde.org/) 3) Zinf. (former Freeamp). (http://www.zinf.org/) (http://www.freeamp.org/) 4) Gqmpeg. (http://gqmpeg.sourceforge.net/) 5) SnackAmp.(http://snackamp.sourceforge.net/) s)
Reproductores de CD	CD player	1) KsCD. (ttp://r.zorbla.de/milliByte/index.html) 2) Orpheus. (console) (http://thekonst.net/en/orpheus) 3) Sadp. (http://www.geocities.com/xsadp/) 4) WorkMan. (http://ftp.x.org/contrib/applications/WorkMan/) 5) Xmcd. (http://www.amb.org/xmcd/) 6) Grip. (http://www.nostatic.org/grip/)
Secuenciador de Midi	Cakewalk	1) RoseGarden. (http://www.rosegardenmusic.com/) 2) Brahms. (http://brahms.sourceforge.net/) 3) Anthem. (http://anthem.sourceforge.net/) 4) Melys. (http://www.parabola.demon.co.uk/melys/) 5) MuSE. (http://www.muse-sequencer.org/)Más información: Midi-Howto. (http://www.tldp.org/HOWTO/MIDI-HOWTO-8.html)
Trabajando con protocolos en Tiempo Real	RealPlayer	1) RealPlayer for Linux.(http://www.real.com/linux/) 2) Mplayer + librerías. (http://www.mplayerhq.hu/design7/news-es.html)

3) Multimedia (Gráficos)

Descripcion del programa, tareas ejecutadas	Windows	Linux
Convertidores de video	Virtual Dub	1) Transcode. (http://www.transcoding.org/cgi-bin/transcode)
Creación de Flash	Macromedia Flash	1) DrawSWF. (http://drawswf.sourceforge.net/index.html) 2) Ming. (Creación de flash swf) (http://ming.sourceforge.net/)
Creación de gráficos Web	Macromedia Fireworks	Gimp. (http://www.gimp.org/)
Creación y edición Profesional de Video	Adobe Premiere, Media Studio Pro	1) Cinelerra. (http://heroinewarrior.com/cinelerra.php3) 2) MainActor.(http://mainactor.softonic.com/linux) Es una aplicación multiplataforma para la edición no lineal de vídeo digital. 3) Broadcast 2000. 4) Lives. (http://lives.sourceforge.net/)
Creación y edición sencilla de video	Windows Movie Maker	1) MainActor.(http://mainactor.softonic.com/linux) Es una aplicación multiplataforma para la edición no lineal de vídeo digital. 2) Broadcast 2000.
Dibujo de Diagramas y Estructuras de Bases de Datos	Access	1) Dia. (http://www.gnome.org/projects/dia/) 2) Toolkit for Conceptual Modelling. (http://wwwhome.cs.utwente.nl/~tcm/)
Editores de Iconos	Microangelo	1) Gnome-iconedit. (http://linux.about.com/cs/linux101/g/gnomeiconedit.htm) Es un editor de iconos para GNOME. 2) Kiconedit. (http://docs.kde.org/development/en/kdegraphics/kiconedit/) Está diseñado para asistir en la creación de iconos para KDE .
Editores potentes de Gráficos tipo PhotoShop	Adobe Photoshop, Paint Shop Pro	1) Gimp, FilmGimp. (http://www.gimp.org/)(http://www.cinepaint.org/) 2) ImageMagick. (http://www.imagemagick.org/script/index.php)
Editores Simples de Gráficos	Paint	1) Tuxpaint. (http://www.tuxpaint.org/) 2) Gpaint. (http://www.gnu.org/software/gpaint/)
Gráficos 3D	3D Studio MAX, Maya, Povray, ...	1) Blender. (http://www.blender.org/) 2) Maya. (http://www.highend3d.com/maya/) Es un software para la creación de gráficos 3D que permite efectos especiales y de animación 3)KPovModeler. (http://www.kpovmodeler.org/) 4) K3Studio. (http://k3studio.sourceforge.net/) 5) Moonlight. (http://www.moonlight3d.eu/cms/) 6) GIG3DGO. (http://meanwhile.com/?domain=gig3d.com&) 7) Povray. (http://www.povray.org/) 8) K3D. (http://www.k-3d.org/wiki/Main_Page) 9) Wings 3D. (http://www.wings3d.com/)
Pequeños programas de captura de pantallas	Integrado con el sistema(PrintScreen), Snag it, ...	1) Ksnapshot. (http://ksnapshot.softonic.com/linux) Es una que le permite tomar capturas de pantalla . Es capaz de capturar imágenes de todo el escritorio, de una sola ventana, de parte de una ventana o del área seleccionada. 2) Xwpick.(http://hpux.connect.org.uk/hppd/hpux/X11/Graphics/xwpick-2.20/) Programa de captura de pantallas 3) Xwd, xgrabsc. Programa de captura de pantallas

Descripcion del programa, tareas ejecutadas	Windows	Linux
Programa para Decorar Texto	Wordart	OpenOffice Draw. (http://www.openoffice.org/product/draw.html)
Programas para Trabajos gráfico con Vectores	Adobe Illustrator, Corel Draw 14, Freehand, AutoSketch	1) Sodipodi. (http://www.sodipodi.com/index.php3) 2) xfig. (http://www.xfig.org/) 3) Sketch. (http://www.skencil.org/) 4) Karbon14 and Kontour. (http://www.koffice.org/karbon/) 5) OpenOffice Draw. (http://www.openoffice.org/) 6) Dia. (http://www.gnome.org/projects/dia/) 7) Tgif. (http://bourbon.usc.edu/tgif/) 8) ImPress. (http://www.ntlug.org/~ccox/impress/)
Reproductores de DVD	PowerDVD, WinDVD, MicroDVD, Windows Media Player, VideoLAN	1) Ogle. (http://www.dtek.chalmers.se/groups/dvd/) 2) Mplayer. (http://www.mplayerhq.hu/design7/news.html) 3) Xine. (http://xinehq.de/) 4) VideoLAN. (http://www.videolan.org/)
Reproductores de Video / mpeg4	BSplayer, Zoomplayer, Windows Media Player, VideoLAN	1) Mplayer. (http://www.mplayerhq.hu/design7/news.html) 2) Xine. (http://xinehq.de/) 3) Sinek. (frontend para libxine) (http://sinek.sourceforge.net/) 4) VideoLAN. (http://www.videolan.org/)
Trabajando con Sintonizadores de TV	AVerTV, PowerVCR 3.0, CinePlayer DVR,	1) Xawtv. (http://linux.bytesex.org/xawtv/) 2) GnomeTV.(http://gnometv.sourceforge.net/) 3) Mplayer. (http://www.mplayerhq.hu/design7/news.html)
Visualizador de Archivos Gráficos	ACDSee, IrfanView	1) Xnview. (http://www.xnview.com/) 2) Gqview. (http://gqview.sourceforge.net/) 3) Qiv. (http://www.klografx.net/qiv/) 4) Kuickshow. (http://kuickshow.sourceforge.net/) 5) xv. [Prop] (http://www.trilon.com/xv/xv.html) 6) imgv. (http://www.seekrut.com/) 7) Gwenview.(http://gwenview.sourceforge.net/) 8) Showimg. (http://www.jalix.org/projects/showimg/) 9) Gthumb. (http://gthumb.sourceforge.net/)

4) Ofimática/Negocios

Administrador de Proyectos	MS Project, Project Expert 7	Mr Project. (http://www.gnome.org/gnome-office/mrproject.shtml)
Administrador Personal de Finanzas	MS Money, Quicken	1) GNUcash. (http://www.gnucash.org/es/) 2) GnoFin. (http://gnofin.sourceforge.net/) 3) Kmymoney. (http://kmymoney2.sourceforge.net/index-home.html) 4) Grisbi. (http://www.grisbi.org/)
Paquetes de oficina	MS Office	1) Openoffice. (http://www.openoffice.org/) 2 Koffice. (http://www.koffice.org/) 3) Gnome Office. (http://www.gnome.org/gnome-office/) 4) Applixware Office.(http://www.vistasource.com/vs2/en/index.php) 5) LaTeX, ... (http://www.latex-project.org/)

5) Para correr un sistema sin instalar

Descripcion del programa, tareas ejecutadas	Windows	Linux
Sistemas, corriendo desde un CD sin instalarse (Live CD)	Imposible	8) SPEC SFS. [Prop] (http://www.spec.org/osg/sfs97r1/) 1) Knoppix. (http://www.knopper.net/knoppix/index-en.html) 2) Ubuntu http://www.ubuntu.com/ 3) Kubuntu http://www.kubuntu-es.org/ 4) Cool Linux. (http://sourceforge.net/projects/emergencycd2/) 5) DemoLinux. (http://www.demolinux.org/) 6) DyneBolic. (http://lab.dyne.org/DyneBolic) 7) Gentoo (live CD). (http://www.gentoo.org/) 8) Lonix. (http://lonix.sourceforge.net/es/index.html) 9) ByzantineOS. (http://byzgl.sourceforge.net/wiki/index.php/Main_Page) 10) MoviX. (http://sourceforge.net/projects/movix/) 11) SuSE live-eval CD.(http://www.novell.com/es-es/linux/) 12) Eagle Linux. (http://www.geocities.com/eaglelinux/)

6) Redes y Conectividad

Administración Remota	VNC, Administrador Remoto, Asistencia Remota, Symantec pcAnywhere,	1) VNC. (http://www.cl.cam.ac.uk/research/dtg/attarchive/vnc/index.html) Nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. 2) ssh / openssh. (http://www.ssh.com/) (http://www.openssh.org/). Proporciona un método para usar encriptación externa entre cliente y servidor. El protocolo SSH es un método general de encriptación, y puede ser aplicado de forma general para cualquier aplicación. 3) Remote management is built-in in Xfree86. (http://www.xfree86.org/). Provee una interfaz gráfica cliente/servidor entre el hardware (ratón, teclado y sistemas gráficos) y un entorno de escritorio que provee un sistema de ventanas así como una interfaz estandarizada de aplicación (API por sus siglas en inglés). 4) Rdesktop Client. (http://www.rdesktop.org/)
Clientes de Email	Outlook Express, Eudora	1) Thunderbird (http://www.mozilla-europe.org/es/products/thunderbird/). Aplicación de correo electrónico 2) Evolution (http://www.gnome.org/projects/evolution/) Es un gestor libre de información personal y de trabajo en grupo . Combina administración de correo electrónico, calendario, agenda y lista de tareas. 3) Kmail.(http://kontact.kde.org/). Es una herramienta de correo electrónico para KDE, el entorno de escritorio K. Dispone de una interfaz gráfica intuitiva que facilita el uso.
Clientes para Mensajería Instantánea	ICQ, MSN, AIM, ...	1) Kopete. (http://kopete.kde.org/) Kopete es una aplicación informática libre para mensajería instantánea flexible y sistema multiprotocolo extendible 2) Pidgin (Anteriormente GAIM) Cliente multiplataforma de mensajería instantánea. http://pidgin.im/pidgin/home/ http://www.jabberes.org/clientes/gaim/

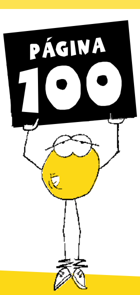
Descripcion del programa, tareas ejecutadas	Windows	Linux
Compartiendo archivos clientes / servidor (redes punto a punto)	Morpheus (Gnutella), WinMX	1) LimeWire for Linux. (Gnutella). (http://www.limewire.com/english/content/home.shtml). Cliente para el intercambio de archivos P2P que busca principalmente archivos MP3 y películas en la red Gnutella. También puedes iniciar búsquedas de cualquier tipo de archivo (fotos, juegos, vídeos). 2) Lopster. (OpenNAP) (http://lopster.sourceforge.net/) Es un cliente Napster para X11 escrito en GTK. Actualmente es el cliente gráfico más completo existente para linux. 3) Mldonkey. (eDonkey) (http://mldonkey.sourceforge.net/Main_Page). Es un programa P2P multired capaz de conectarse a varias redes simultáneamente, y descargar un solo fichero de varias redes a la vez. 4) Bittorrent. (http://www.bittorrent.com/)
Compartiendo Datos/Archivos	Windows shares	1) Samba. (http://us3.samba.org/samba/) 2) Samba-TNG. (http://www.samba-tng.org/)
Comunicación por voz	Speak Freely	1) Speak Freely for Unix. (http://speak-freely.sourceforge.net/) 2) TeamSpeak. (http://www.goteamspeak.com/). Es un servidor de voz chat.
Conferencias en línea con Video/audio	NetMeeting	GnomeMeeting. (http://www.gnomemeeting.org/). Es una aplicación para realizar videoconferencias y telefonía por IP para GNOME.
Contabilidad de tráfico	Tmeter, ...	1) Getstatd. (http://www.opennet.ru/getstatd/) 2) Ipac-ng. (http://sourceforge.net/projects/ipac-ng) Herramienta para realizar estadísticas sobre tráfico IP. legato Copias de seguridad. 3) Ipaudit. (http://ipaudit.sourceforge.net/ipaudit-web/). Da estadísticas y crea informes de trafico existente entre direcciones IP
Gestor de Descargas	Flashget, Getright	1) Prozilla. (http://prozilla.genesys.ro/) Es un completo acelerador de descargas y es un programa esencial para cualquier usuario que realice descargas frecuentemente ya que le reducirá al máximo el tiempo dedicado a éstas. 2) DownloadThemAll (Extensión para Firefox)
Mensajería local con máquinas bajo Windows	WinPopUp	1)LinPopUp.(http://web.telia.com/~u89404340/linpopup/) Programa de mensajería instantánea que permite enviar y recibir mensajes a clientes Winpopup o RealPopup, haciendo uso de Samba. 2) Kpopup. (http://www.henschelsoft.de/kpopup_en.html) Kpopup es un programa que permite intercambiar mensajes con el programa Winpopup y otros compatibles (como Linpopup).
Monitoreo de web Sites o Bandejas de Mail, muestran los titulares o todo el artículo	WatzNew	1) Web Secretary. (http://savannah.nongnu.org/projects/websec/)

6) Redes y Conectividad

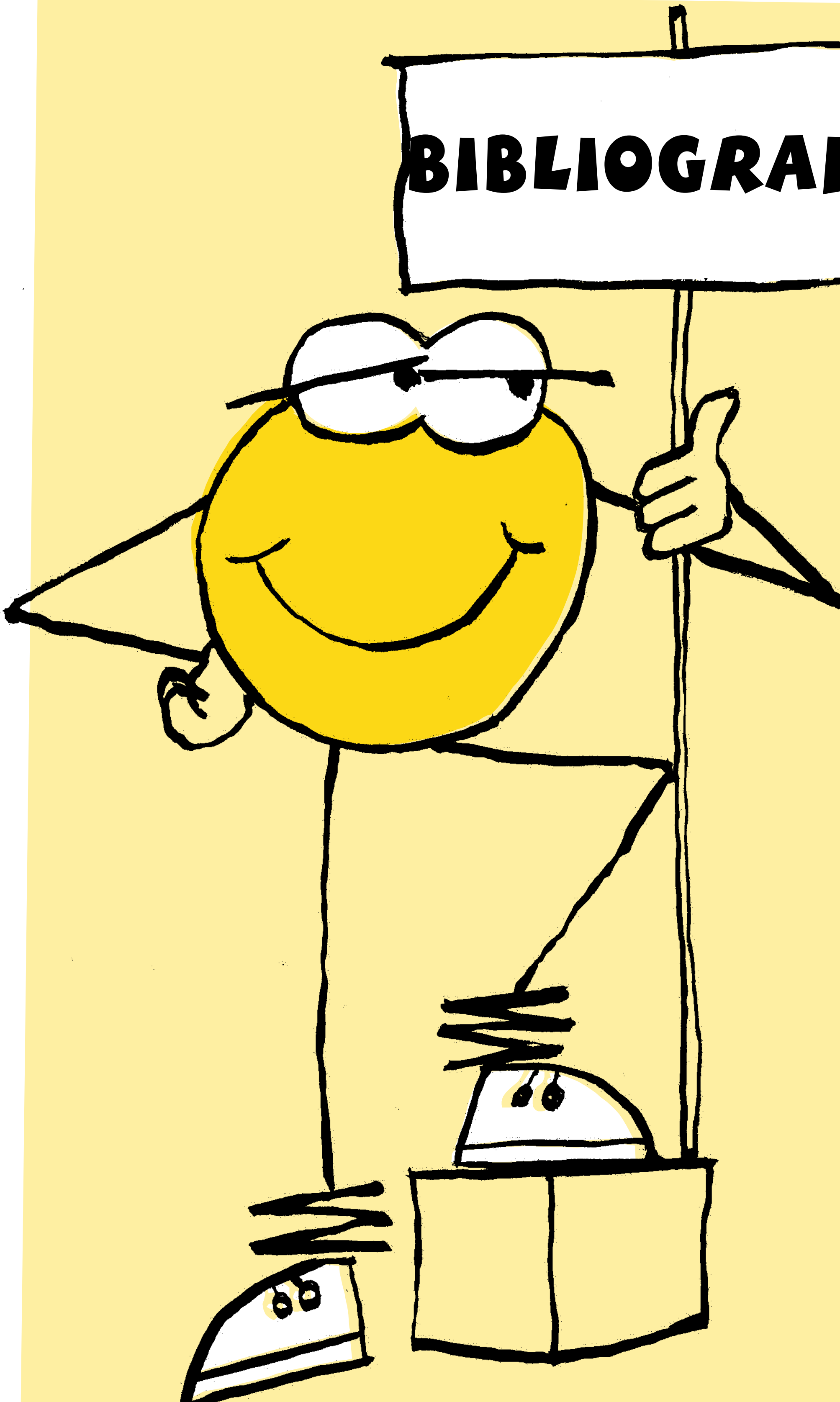
Descripcion del programa, tareas ejecutadas	Windows	Linux
Navegadores Web	Internet Explorer	1) Firefox (http://www.mozilla-europe.org/es/) 2) Galeon (http://galeon.sourceforge.net/) Es un navegador libre creado para el proyecto GNOME. 3) Konqueror (http://www.konqueror.org/) Es un navegador libre y parte oficial del proyecto KDE. Funciona como visor y gestor de archivos, y también como navegador web.
Trabajando con Faxes	WinFax	1) HylaFax. (http://www.hylafax.org/content/Main_Page) 2) Fax2Send. (http://www.fax2send.com/fax2send/page/home.html). Es un programa servidor para poder enviar y recibir los faxes .

7) Sistemas de Software para Escritorio

Creador de PDF	Adobe Acrobat Distiller	1) Open Office (http://es.openoffice.org) 2) Xfig. (http://epb.lbl.gov/xfig/) Es una completa herramienta de dibujo, con ella podemos realizar cualquier tipo de dibujo vectorial en Linux; gráficos, planos, esquemas, bocetos, diagramas de flujo. 3) Ghostscript (http://www.cs.wisc.edu/~ghost/) 4) Tex2Pdf. (http://tex2pdf.berlios.de/) 5) GV. (http://wino.physik.uni-mainz.de/~plass/gv/)
Criptografía y Cifrado	PGP, GnuPG + Windows Privacy Tools	1) GnuPG (console) + GPA, KGpg. (http://www.gnupg.org/) (http://www.gnupg.org/(en)/related_software/gpa/index.html) (http://developer.kde.org/~kgpg/) (http://www.gnupg.org/(en)/related_software/frontends.html#gui)
Editor de Consola para Texto	FAR Editor, ...	1) Vim. (http://www.vim.org/) 2) EMACS http://www.gnu.org/software/emacs/emacs.html
Editor de Texto	Notepad, WordPad, TextPad,...	1) Kedit (KDE). 2) Gedit (Gnome). 3) Gnotepad. (http://gnotepad.sourceforge.net/) 4) Kate (KDE). (http://www.kate-editor.org/) 5) KWrite (KDE). 6) Vim. (http://www.vim.org/)
Editor de texto y código multifinalidad	SciTE, UltraEdit, MultiEdit,...	1) Kate (KDE). (http://www.kate-editor.org/) 2) Nedit. (http://www.nedit.org/) 3) CodeCommander. (http://glimmer.sourceforge.net/) 4) SciTE. (http://www.scintilla.org/SciTE.html) 5) Quanta Plus. (http://quanta.kdewebdev.org/) 6) Vim. (http://www.vim.org/) 7) Xemacs. (http://www.xemacs.org/)
Reconocimiento de Texto (OCR)	Recognita, FineReader	1) ClaraOcr. (http://www.gnupg.org/(en)/related_software/frontends.html#gui) 2) Gocr. (http://jocr.sourceforge.net/)



Descripcion del programa, tareas ejecutadas	Windows	Linux
Reconocimiento de texto por voz	MS text to speech	1) Festival. (http://www.cstr.ed.ac.uk/projects/festival/) 2)Emacspeak.(http://www.cs.cornell.edu/Info/People/raman/emacspeak/emacspeak.html)
Trabajando con compresores de archivos	WinZip, WinRar	1) FileRoller. (http://fileroller.sourceforge.net/) 2) Gnozip. (http://www.geocities.com/SiliconValley/9757/gnozip.html) 3) LinZip. (http://linzip.softonic.com/linux) 4) Ark (kdeutils). 5) Karchiveur. (http://perso.orange.fr/coquelle/karchiver/) 6) Gnochive. (http://gnochive.sourceforge.net/) 7) RAR for Linux. (http://www.rarlab.com/download.htm)
Visualizador de PDF	Adobe Acrobat Reader	1) Acrobat Reader para Linux. (http://www.adobe.com/products/acrobat/) 2) Xpdf. (http://www.foolabs.com/xpdf/)
Visualizador de PostScript	RoPS	1) GhostView. (http://www.cs.wisc.edu/~ghost/) 2) Kghostview. (http://docs.kde.org/development/en/kdegraphics/kghostview/)
8) Trabajando con Archivos		
Administrador de Archivos al estilo FAR y NC	FAR, Norton Commander, etc	1) Midnight Commander. (http://www.ibiblio.org/mc/) 2) X Northern Captain. (http://xnc.dubna.su/) 3) Deco (Demos Commander) . (http://sourceforge.net/projects/deco/) 4) Portos Commander. (http://pcmm.sourceforge.net/) 5) Konqueror in MC style. (http://www.konqueror.org/)
Administrador de Archivos al estilo Windows	Windows Explorer	1) Konqueror. (http://www.konqueror.org/) 2) Gnome-Commander. (http://www.gnome.org/) 3) Nautilus. (http://www.gnome.org/projects/nautilus/)
Administrador de Archivos al estilo Windows Commander	Total Commander (former Windows Commander)	1) Midnight Commander. (http://www.ibiblio.org/mc/) 2) Krusader. (http://krusader.sourceforge.net/) 3) Kcommander. (http://www.kcommander.org/)
Inspección rápida de documentos de HTML locales	Internet Explorer	1) Konqueror. ((http://www.konqueror.org/) 2) Nautilus. (http://www.gnome.org/projects/nautilus/) 3) Lynx / Links. (http://lynx.browser.org/) (http://links.twibright.com/)
Visualizador de todas las Operaciones con Archivos	Filemon	Filemon para Linux. (http://dar.linux.free.fr/)


BIBLIOGRAFÍA



BIBLIOGRAFÍA

BERNERS-LEE, Tim. El inventor de la World Wide Web, nos descubre su origen. Madrid. Siglo XXI de España Editores. 2000.

DALHEIMER, Matthias; WELSH, Matt. Guía de referencia y aprendizaje Linux. 5a ed. Madrid. Ediciones Anaya Multimedia. 2006.

FINK, M. The business and economics of Linux and open source . Upper Saddle River : Prentice Hall PTR, 2003.

RAYMOND, Eric S. The cathedral and the bazaar : musings on Linux and Open Source by an accidental revolutionary. Rev. ed. Beijing; Cambridge, Mass. : O'Reilly, 2001. 241 p.

RÍOS RUIZ, Wilson Rafael. Aspectos legales del software libre o de Código Abierto (Open Source). En: Derecho de Internet y telecomunicaciones. Grupo de estudios en "Internet, comercio electrónico & telecomunicaciones e informática" (Colombia). Bogotá: Legis, 2003.

SPINELLIS, D. Code reading : the open source perspective . Boston: Addison-Wesley, 2003.

STALLMAN, Richard M. Software libre para una sociedad libre. 1a ed. Madrid: Traficantes de Sueños, 2004. 317 p.

WILLIAMS, S. Free as in freedom : Richard Stallman's crusade for free software. Beijing : O'Reilly, 2002.

ZARZA VILLAGRÁN, Miguel Angel. ¿Qué es el movimiento de software libre? Regulación y autoorganización. En: Internet y pluralismo jurídico: formas emergentes de regulación. Pompeu Casanovas, ed. Granada: Comares, 2003.

Documentos electrónicos disponibles en Internet

First Monday Special Issue #2: Open Source October 2005 [En línea] Disponible en: http://firstmonday.org/issues/special10_10/ [fecha de consulta: 19 de Junio de 2007]

FREE SOFTWARE FUNDATION. Nosotros hablamos de software libre [En línea]. s.l: fsfeurope.org, 2005 [Fecha de consulta: 23 de junio de 2007]. Disponible En: http://www.fsfeurope.org/about/about.es.html

FSF - The Free Software Foundation [En línea] Disponible en: http://www.fsf.org/ [Fecha de consulta: 19 de Junio de 2007]

Fundación Vía Libre [En línea] Disponible en: http://www.vialibre.org.ar [fecha de consulta: 19 de Junio de 2007]

GONZÁLEZ BARAHONA, Jesús; SEOANE PASCUAL, Joaquín; ROBLES, Gregorio. Introducción al Software Libre. [En línea] Disponible en: http://www.uoc.edu/masters/cat/img/693.pdf [fecha de consulta: 22 de junio de 2007].

La Pastilla Roja [En línea] Disponible en: http://www.lapastillaroja.net/ [fecha de consulta: 19 de Junio de 2007]

Libro blanco del Software Libre en España [En línea] Disponible en: http://libroblanco.com/joomla/ [fecha de consulta: 19 de Junio de 2007]

MAS I HERNÁNDEZ, Jordi. Software libre en el sector público [en línea]. S.l.: uoc.edu, 2003 [fecha de consulta: 23 de junio de 2007]. Disponible en: < http://www.uoc.edu/dt/20327/>

Novática [en línea]. Barcelona : Asociación de Técnicos en Informática, 1975. 2005, núm. 175 : El software libre como objeto de estudio. [Fecha de consulta : 23/06/07]. Disponible en < http://www.ati.es/novatica/2005/175/nv175sum.html> 2003, núm. 163 : Conocimiento abierto (open knowledge). [Fecha de consulta . 23/06/07]. Disponible en http://www.ati.es/novatica/2003/163/ nv163sum.html 2001, núm. 154 : Software libre : fuente abierta hacia la madurez. [Fecha de consulta : 23/06/07]. Disponible en http://www.ati.es/novatica/2001/154/nv154sum.html

OFSET - Organization for Free Software in Education and Teaching [En línea] Disponible en: http://www.ofset.org/ [Fecha de consulta: 20 de Junio de 2007]

RAYMOND, S. Eric. La Catedral y el Bazar – Wikisource [En línea] Disponible en: <http://es.wikisource.org/wiki/La_Catedral_y_el_Bazar>[fecha de consulta: 22 de Junio de 2007] Optaros Open Source Catalogue 2007 [En línea] Disponible en: http://www.optaros.com/en/publications/white_papers_reports/open_source_catalogue_2007 [fecha de consulta: 19 de Junio de 2007]

REFERENCIAS: software libre en los Estados del mundo [En línea]. Chile: softwarelibre.cl, 2002. Disponible en: < http://www.softwarelibre.cl/drupal/?q=referencias> [Fecha de consulta: 22 de junio de 2007]

STALLMAN, Richard. Software libre para una sociedad libre. [en línea]. Disponible en: <http://www.traficantes.net/index.php/trafis/editorial/catalogo/coleccion_mapas/software_libre_para_una_sociedad_libre> [fecha de consulta: 22 de junio de 2007].

The IDA Open Source Migration Guidelines [en línea] Disponible en:
http://www.netproject.com/docs/migoss/ [Fecha de consulta: 22 de junio de 2007]¨

UNESCO y el Software Libre [en línea] Disponible en:
http://www.unesco.org.uy/informatica/consorcio/stallman.html
[Fecha de consulta: 12 de Octubre de 2007]

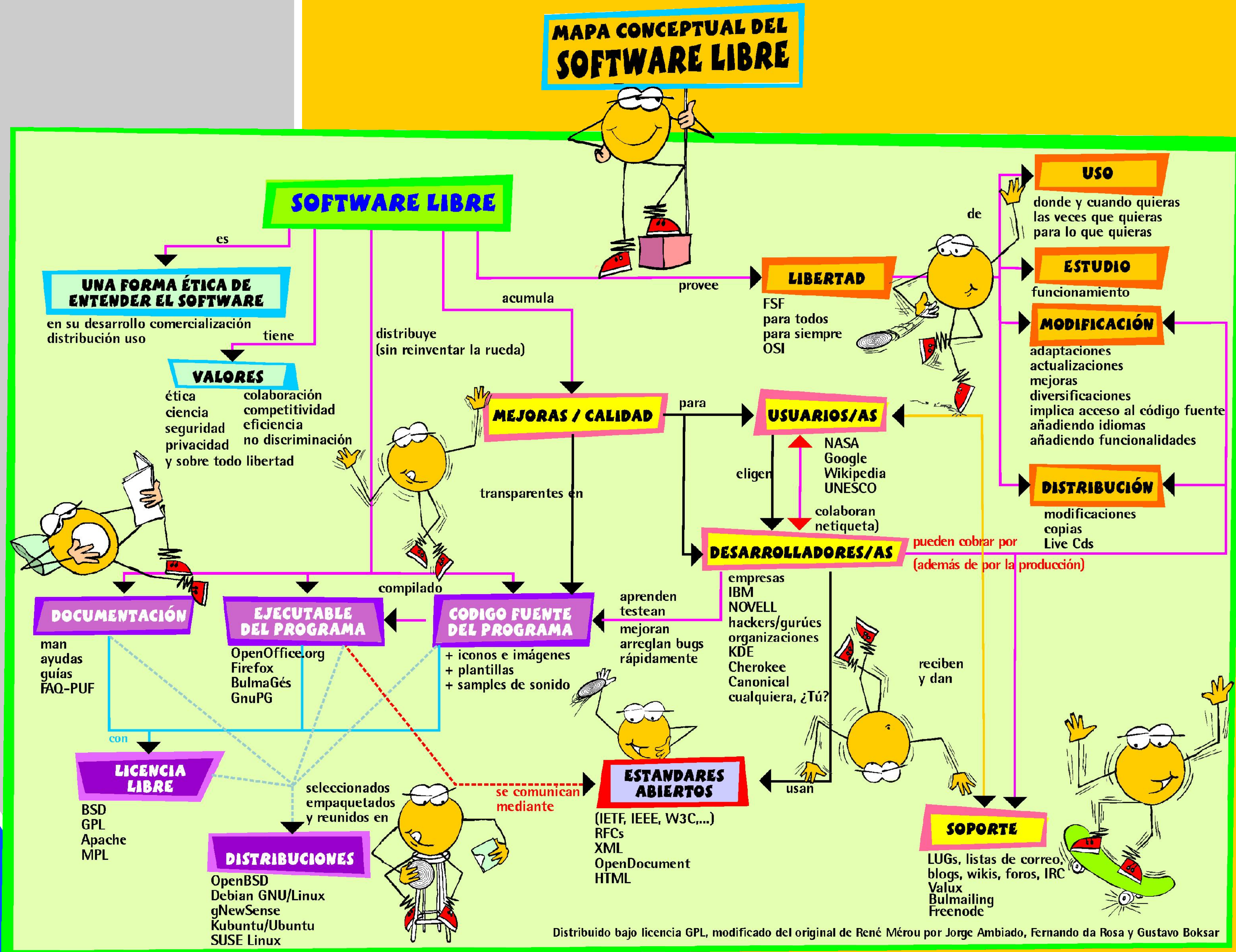
Unofficial Translations - GNU Project - Free Software Foundation (FSF) [En línea] Disponible en:
http://www.gnu.org/licenses/translations.html [Fecha de consulta: 19 de Junio de 2007]

Bibliografías complementarias

Free/Open Source Software Academic Bibliography [en línea] Disponible en: http://wwwai.wu-wien.ac.at/%7Ekoch/forschung/sw-eng/oss_list.html [Fecha de consulta: Martes, 19 de Junio de 2007]

Open Source Software and Libraries Bibliography [en línea] Disponible en:
<http://www.vuw.ac.nz/staff/brenda_chawner/biblio.html> [fecha de consulta: Martes, 19 de Junio de 2007]

Women in Open Source/Free Software bibliography | LinuxChix [en línea] Disponible en:
http://www.linuxchix.org/women-open-source-free-software-bibliography.html [fecha de consulta: Martes, 19 de Junio de 2007]


MAPA CONCEPTUAL DEL SOFTWARE LIBRE
SOFTWARE LIBRE
es
UNA FORMA ÉTICA DE ENTENDER EL SOFTWARE
en su desarrollo comercialización distribución uso
tiene
VALORES
ética
ciencia
seguridad
privacidad
y sobre todo libertad
colaboración
competitividad
eficiencia
no discriminación
distribuye (sin reinventar la rueda)
acumula
provee
LIBERTAD
FSF
para todos
para siempre
OSI
de
USO
donde y cuando quieras
las veces que quieras
para lo que quieras
ESTUDIO
funcionamiento
MODIFICACIÓN
adaptaciones
actualizaciones
mejoras
diversificaciones
implica acceso al código fuente
añadiendo idiomas
añadiendo funcionalidades
DISTRIBUCIÓN
modificaciones
copias
Live Cds
MEJORAS / CALIDAD
para
USUARIOS/AS
NASA
Google
Wikipedia
UNESCO
eligen
colaboran (netiqueta)
DESARROLLADORES/AS
pueden cobrar por (además de por la producción)
empresas
IBM
NOVELL
hackers/gurúes
organizaciones
KDE
Cherokee
Canonical
cualquiera, ¿Tú?
transparentes en
compilado
aprenden
testean
mejoran
arreglan bugs
rápidamente
DOCUMENTACIÓN
man
ayudas
guías
FAQ-PUF
EJECUTABLE DEL PROGRAMA
OpenOffice.org
Firefox
BulmaGés
GnuPG
CODIGO FUENTE DEL PROGRAMA
+ iconos e imágenes
+ plantillas
+ samples de sonido
con
LICENCIA LIBRE
BSD
GPL
Apache
MPL
seleccionados empaquetados y reunidos en
DISTRIBUCIONES
OpenBSD
Debian GNU/Linux
gNewSense
Kubuntu/Ubuntu
SUSE Linux
se comunican mediante
ESTANDARES ABIERTOS
(IETF, IEEE, W3C,...)
RFCs
XML
OpenDocument
HTML
usan
reciben y dan
SOPORTE
LUGs, listas de correo, blogs, wikis, foros, IRC
Valux
Bulmailing
Freenode
Distribuido bajo licencia GPL, modificado del original de René Mérou por Jorge Ambiado, Fernando da Rosa y Gustavo Boksar